electronics

Article

A Novel Traffic Obfuscation Technology for Smart Home

Shuo Zhang, Fangyu Shen, Yaping Liu *, Zhikai Yang and Xinyu Lv

Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China; szhang18@gzhu.edu.cn (S.Z.); 2112006189@e.gzhu.edu.cn (F.S.); 2112006047@e.gzhu.edu.cn (Z.Y.); 2112006171@e.gzhu.edu.cn (X.L.)

* Correspondence: ypliu@gzhu.edu.cn

Abstract: With the widespread popularity of smart home devices and the emergence of smart home integration platforms such as Google, Amazon, and Xiaomi, the smart home industry is in a stage of vigorous development. While smart homes provide users with convenient and intelligent living, the problem of smart home devices leaking user privacy has become increasingly prominent. Smart home devices give users the ability to remotely control home devices, but they also reflect user home activities in traffic data, which brings the risk of privacy leaks. Potential attackers can use traffic classification technology to analyze traffic characteristics during traffic transmission (e.g., at the traffic exit of a smart home gateway) and infer users' private information, such as their home activities, causing serious consequences of privacy leaks. To address the above problems, this paper focuses on research on privacy protection technology based on traffic obfuscation. By using traffic obfuscation technology to obscure the true traffic of smart home devices, it can prevent malicious traffic listeners from analyzing user privacy information based on traffic characteristics. We propose an enhanced smart home traffic obfuscation method called SHTObfuscator (Smart Home Traffic Obfuscator) based on the virtual user technology concept and a virtual user behavior construction method based on logical integrity. By injecting traffic fingerprints of different device activities into the real traffic environment of smart homes as obfuscating traffic, attackers cannot distinguish between the real device working status and user behavior privacy in the current home, effectively reducing the effect of traffic classification attack models. The protection level can be manually or automatically adjusted, achieving a balance between privacy protection and bandwidth overhead. The experimental results show that under the highest obfuscation level, the obfuscation method proposed in this paper can effectively reduce the classification effect of the attack model from 95% to 25%.

Keywords: smart home privacy; traffic obfuscation; traffic fingerprint

check for updates

Citation: Zhang, S.; Shen, F.; Liu, Y.; Yang, Z.; Lv, X. A Novel Traffic Obfuscation Technology for Smart Home. *Electronics* **2023**, *12*, 3477. https://doi.org/10.3390/electronics12163477

Academic Editors: Daniele Riboni, Athanasios D. Panagopoulos and Myung-Sup Kim

Received: 21 June 2023
Revised: 31 July 2023
Accepted: 7 August 2023
Published: 17 August 2023

1. Introduction

In recent years, the smart home industry is in a rapid development stage. According to research conducted by relevant institutions, it is predicted that by 2025, 21.3% of households worldwide will use smart home devices, and the total number of smart home devices will reach 5.44 billion units [1]. While smart home devices provide users with convenient and intelligent living experiences, they also pose privacy risks by reflecting users' home activities in traffic data. Smart home devices are typically limited in functionality and designed for specific purposes. The changes in traffic patterns are highly correlated with user behavioral activities. Potential attackers can exploit this by performing traffic classification attacks to identify the activity states of users' devices, as illustrated in Figure 1, particularly in the traffic exit of smart home gateways. Therefore, it is relatively easy to identify user activities and infer privacy information from smart home traffic data, leading to significant privacy risks in this regard.

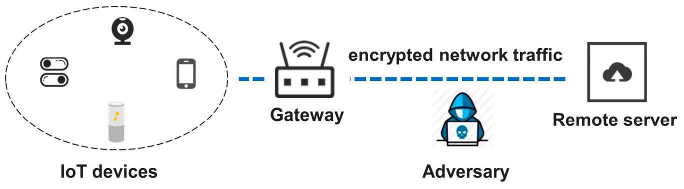

Figure 1. Overview of the privacy threat model.

Relevant studies [2] have shown that even with data encryption, attackers can infer users' privacy-sensitive information from traffic data, as each device has unique traffic characteristics and corresponding activity events. Information such as device types, actions, statuses, and user behaviors can be deduced from the traffic, making it possible to identify user activity even during interactions with smart home devices, such as conversing with voice assistants, opening/closing smart door locks, or watching smart TVs. Detecting changes in traffic rates, averages, and other features can reveal user behavioral patterns and activity states. In specific cases, even the traffic variations in a smart light can provide insights into a household's daily routines, including when users come home, when children go to bed, or whether a user stays up late. Such privacy-sensitive information can potentially facilitate further attacks, such as break-ins when no one is home, crimes targeting individuals living alone, or the sale of personal information to criminals for profit.

Previous traffic classification techniques for smart home traffic have primarily relied on the concept of traffic fingerprints according to [3–6]. Based on relevant information about smart home devices and their operational traffic data, traffic data are transformed into representative feature vectors. These feature vectors are used to construct traffic fingerprints for each device and even for specific device behaviors within the target household. By matching and detecting traffic fingerprints, the behavior of devices can be identified, and user privacy can be inferred.

In this paper, we focus on the side channel attacks where adversaries monitor the incoming/outgoing network traffic to/from smart homes, infer activities of smart home devices with traffic classification algorithms, and gain advantages in conducting subsequent severe attacks. We propose the use of traffic obfuscation techniques at the traffic exit of smart home gateways to obfuscate the actual traffic generated by devices, thereby preventing attackers from inferring users' privacy information based on traffic characteristics. To identify the appropriate timing for traffic injection, we adopted the concept of virtual users. The concept of virtual users in this research is inclined towards constructing a logically coherent and realistic user behavior pattern that is projected onto the household traffic, rather than a combination of individual or a few device behaviors. This approach aims to ensure that the injected false traffic is more deceptive, thereby preventing attackers from extracting genuine user privacy information. The key challenge lies in developing an adaptive strategy for generating virtual user behaviors.

Furthermore, considering the cost and bandwidth constraints in real smart home environments, another significant research focus in recent years has been on achieving a defense mechanism that can be adjusted based on the current network conditions. For example, one approach proposes adjusting the intensity of noise addition by altering the privacy budget [7], allowing users to select an appropriate defense level according to their specific circumstances. This addresses the practicality of implementing effective defense measures while considering the limitations of smart home environments.

Overall, we propose an enhanced smart home traffic obfuscation method called SHTObfuscator (Smart Home Traffic Obfuscator). The method utilizes the concept of virtual users and injects obfuscated traffic to construct a "realistic" user behavior trajectory within the smart home traffic data. Additionally, a smart home traffic privacy protection system, SHTProtector (Smart Home Traffic Protector), is designed and implemented. The functionality of each module in the system is tested through experiments, validating the feasibility and effectiveness of the SHTObfuscator approach. The specific research work is as follows:

The contribution of this paper is as follows:

(1) The proposal of an enhanced smart home traffic obfuscation method called SHTObfuscator based on the virtual user technology concept. By injecting traffic fingerprints of different device activities into the real traffic environment, we effectively reduce the effect of traffic classification attack models.
(2) A smart home traffic privacy protection system SHTProtector is designed and implemented. Experiments of device identification monitoring, device fingerprint extraction, traffic obfuscation effect and traffic obfuscation overhead are carried out in the real environment of smart home, and the effectiveness of the proposed method is verified.
(3) Achieved the balance between privacy preserving and communication overhead in accordance with the network condition.

The organizational structure of this paper is as follows. We introduce the related work of traffic obfuscation for smart homes in Section 2, and the motivation in Section 3. We propose the design and simulation of our mechanism in Section 4 and the experimental evaluation in Section 5. We conclude this paper in Section 6.

2. Related Work

Attackers can infer the privacy of smart homes through traffic classification. However, traffic obfuscation techniques can confuse real traffic, thereby preventing attackers from inferring users' private information based on traffic characteristics. Current research on smart home traffic obfuscation techniques, both domestically and internationally, mainly includes packet padding, traffic shaping, fake traffic injection, and virtual user.

Yao ZJ [8] proposed an evaluation framework for assessing the effectiveness of traffic obfuscation methods. The evaluation metrics for traffic obfuscation methods include stealthies, computational overhead, and deployment difficulty.

Stealthies refers to the ability of network traffic to be obfuscated in order to evade detection by observers. Computational overhead refers to the number of resources consumed during the obfuscation process, including computational time, number of computations, and required physical resources. Additionally, the deployment difficulty of obfuscation techniques is also an important factor that affects user experience. Therefore, when selecting suitable traffic obfuscation techniques, it is necessary to consider a comprehensive range of factors, including stealthies, computational overhead, and deployment difficulty.

In order to minimize privacy breaches in smart homes, in 2019, Nicolazzo S et al. [9] proposed a privacy-preserving solution for mitigating feature disclosure in a multiple IoT environment, which draws inspiration from concepts in database theory, specifically k-anonymity and t-closeness. Additionally, in 2022, Corradini E et al. [10] proposed a two-tier Blockchain framework to increase the security and autonomy of smart objects in the IoT by implementing a trust-based protection mechanism. They have implemented robust privacy measures to safeguard the personal information of IoT devices.

Packet padding primarily targets attack methods that rely on packet size features, while traffic shaping focuses on overall statistical features of data flows, including temporal patterns, periodicity, and rates of traffic.

In 2018, Pinheiro A J et al. [11] summarized and compared the obfuscation strategies for packet length, and proposed a lightweight packet filling mechanism, which adopted the combination of maximum filling and random filling, and reduced the accuracy of traffic classification algorithm from 92% to 30.38% at the cost of delay increase of approximately 19.8%. The limitation lies in less experimental settings and the need to use VPN.

In 2019, Apthorpe et al. [12] improved and proposed a new traffic shaping algorithm Stochastic Traffic Padding (STP), which allowed users to make a trade-off between cost and privacy. STP performed traffic shaping during user activities and selectively injected confused traffic in other time periods, which improved the disadvantage of constant rate traffic classifier that the cost of no user activities was too high.

Xiong et al. [13] introduced the concept of differential privacy for data packet padding. The authors employed a differential privacy model to obfuscate the traffic of each smart home device. They adjusted the level of obfuscation flexibly by defining different privacy levels, allowing for different obfuscation strengths for low-bandwidth and high-bandwidth devices. This approach aimed to meet the usage requirements of different smart home users.

In 2020, Pinheiro et al. [14] improved upon the static data packet padding mechanism proposed in 2018. To address the high overhead issue, they proposed an adaptive data packet padding method based on Software-Defined Networking (SDN). This method adjusts the number of inserted data packets based on changes in the utilization of the home network. The padding mechanism is set by an SDN application that monitors network traffic variations, with the goal of dynamically balancing privacy protection and communication overhead.

Wang et al. [15] introduced the concept of differential privacy for traffic obfuscation. The authors modeled the distribution of packet inter-arrival times and packet sizes and used privacy parameters to determine the level of privacy protection provided by differential privacy. The advantage of this method is its controllable overhead. However, the drawback is that the obfuscation noise increases the latency of the data packets, which may impact the normal operation of devices.

In 2021, Prates N et al. [16] proposed a defense mechanism that combines active monitoring and passive defense. It includes two modules: a vulnerability monitoring module that collects traffic from smart home devices and extracts requests exchanged between the devices and the gateway along with their timestamps. Possible privacy leaks are inferred using time-based statistical algorithms. The traffic shaping module introduces delays to the traffic of devices experiencing privacy leaks, making it difficult for attackers to perform inference attacks. The limitation of this method is also the lack of in-depth exploration of the impact of delays on the normal operation of devices.

In traffic shaping methods, there are also smart home traffic obfuscation schemes based on adversarial learning, which typically utilize Generative Adversarial Networks (GANs), Deep Convolutional GANs (DCGANs), and similar techniques to generate obfuscation noise for traffic. Relevant research in this area includes:

In 2020, Ibitoye et al. [17] proposed a privacy protection scheme for smart homes based on GANs, specifically addressing the privacy issues of audio devices such as smart speakers. In the face of audio-based inference attacks, this method effectively defends against inference attacks while preserving the semantics of audio samples.

In the same year, Ranieri et al. [18] focused on defense methods against traffic attacks on smart devices, particularly targeting smart speakers. The authors built a testbed for the smart home environment to evaluate the effectiveness of deep adversarial learning techniques. The experiments validated the effectiveness of using Additive White Gaussian Noise (AWGN) for traffic shaping, but the limitation is the lack of further optimization regarding the method's overhead.

Packet padding and traffic shaping aim to blur the characteristics of real traffic by adjusting its features, thus preventing attackers from extracting users' private information from the traffic. On the other hand, the purpose of fake traffic injection is to mimic the traffic generated by real devices by injecting fake traffic, thereby concealing the users' actual activities, and preventing attackers from analyzing their network behavior. These methods have a relatively low deployment difficulty since they do not require modifications at the protocol or device level.

Fake traffic injection can be divided into two types: random injection and adaptive injection. Random injection involves injecting randomly generated fake traffic into the real traffic, making it difficult for attackers to distinguish between genuine and fake traffic. This method can effectively increase the cost of attacks and reduce the success rate but may also increase the false positive rate and potentially impact the normal operation of some smart home devices.

In 2017, Apthorpe et al. [19] explored the feasibility and implementation methods of privacy protection through injecting dummy traffic. The authors suggested that injecting virtual traffic could be done on the device itself or on the smart home gateway. The challenge lies in determining the timing of injection, ensuring that the distribution of the generated fake events is similar to that of real events without creating logical conflicts.

In 2019, Hafeez et al. [20] proposed a method of constructing virtual traffic. To conceal background traffic, they suggested sending a constant stream of traffic on the upstream link regardless of the actual activity of smart home devices. When the devices are inactive, virtual traffic representing device activity is sent on the upstream link, preventing adversaries from identifying the real activities of the smart home devices. The limitation of this method also lies in the increased network overhead.

In 2021, Zhu et al. [21] proposed different traffic injection strategies for devices with different bandwidths. The authors divided smart home devices into two categories: high-bandwidth devices and low-bandwidth devices, allowing for better cost planning. Additionally, the proposed method does not modify the original traffic, which has the advantage of not introducing latency to device traffic and not affecting normal device communication.

In 2022, Xu et al. [22] addressed the attack of traffic classification based on device behavior fingerprints, as proposed by Trimananda et al. [5]. They proposed a low-cost defense method by adding noise to these device behavior fingerprints using mechanisms such as random noise. For example, for a smart outlet, the sequence pattern of its on/off events may have only slight differences, so incorporating them into the same pattern would not incur significant overhead. If the on/off events have the same device behavior fingerprint, attackers would not be able to distinguish between the events occurring on the device.

Virtual users are an optimization method proposed in recent years, based on the concept of fake traffic injection, and utilizing adaptive injection techniques. The limitation of traditional fake traffic injection methods is that attackers can use causal relationship analysis or context integrity detection to infer the occurrence of fake events. This is especially true for random injection methods, which can lead to logical conflicts between the behavior of virtual devices and real devices. Therefore, virtual users are more inclined to construct a logically coherent and realistic user behavior pattern that is projected into the household traffic, rather than a combination of individual or a few device behaviors. This ensures that the injected fake traffic is more deceptive and prevents attackers from prying into the real users' privacy information.

In 2021, Liu et al. [23] proposed a defense mechanism against privacy leakage in smart home wireless networks by constructing virtual users. They improved the deception of virtual users based on real users' behavioral patterns, achieving a good obfuscation effect. However, the limitation of their study is that they only obfuscated the unidirectional traffic from the gateway to the devices, which somewhat weakened the defense effectiveness.

In the same year, Yu et al. [24] presented a low-cost and open-source user defense system targeting user activity attack models based on machine learning and deep learning. The authors employed device behavior fingerprint learning using random forests, user behavior modeling based on LSTM, and device behavior fingerprint injection to obfuscate users' home privacy. The defense system showed promising results but lacked a balance between defense effectiveness and bandwidth consumption.

Based on the above analysis, this study believes that the smart home traffic obfuscation method based on the concept of virtual users has great potential for development. However, further optimization is needed in terms of method design and bandwidth consumption.

3. Motivation

In this section, we first present the threat model, which includes the smart home environment and attackers that we consider. Next, we describe the goals and design challenges.

3.1. Threat Model

Our threat model assumes that the attacker knows which IoT devices are in a home, the identity (e.g., MAC addresses) of the IoT device, and the size and time of the packets each device is sending. We assume all packet payloads are encrypted. The goal of the adversaries is to extract the fingerprint of device events and infer privacy from user events. This goal is intentionally broad to encompass different types of devices. For example, a user event for a sleep monitor might be someone falling asleep or waking up.

Although most packets of smart home devices are encrypted, traffic metadata (e.g., timestamps, lengths, and directions) is still available to attackers. Attackers also have access to unencrypted packet headers, which are used to extract valuable information such as Network and MAC addresses. Via side-channel analysis, attackers can use these data to infer privacy-sensitive information about the target home, such as IoT device types, device states, and user behaviors. For the small number of IoT devices that send unencrypted packets, attackers can see the meaningful information about an IoT device from the payload directly and thus need not use any side-channel analysis method [25]. Countermeasures against information leakage from unencrypted packets are out of the scope of this work.

IoT generates traffic different from traffic generated by other individual devices, such as smartphones, routers, or tablets. Besides, traffic of the IoT network follows a stable pattern and the generated network traffic being very predictable which is different from the traffic in ISPs. Among the many threats to privacy introduced by IoT devices, network traffic classification based on side-channel information, such as packet length, interval between packets, flow direction, and transmission rate, represents a major concern as it can lead to leaks of user data and behavior.

Attackers may sniff the export traffic and extracted the device event fingerprints based on the length packet length with machine learning algorithms. The fingerprints will be used for matching device events and furthermore, inferring user privacy information. For example, the use of the packet size alone in machine learning techniques enables inferring if an individual has sleep disorders or health conditions and/or engages in sexual or extra-marital activities. Along these lines, blackmail and extortion become clear threats that stand to violate and detrimentally affect an individual's autonomy, which is one of the most important aspects of privacy [26]. Therefore, it is essential to provide mechanisms that prevent personal information from IoT devices from being compromised or leaked and potentially used to maliciously make inferences about the private life of individuals.

Based on the analysis mentioned above, the main steps and objectives of the smart home traffic privacy attacker in this study are depicted in Figure 2:

(1) The attacker captures the outgoing traffic from the gateway of the target smart home by sniffing, and uses relevant classification features and algorithms to identify the types of devices present in the household, such as smartphones, computers, and smart home devices such as smart lights, smart cameras, and smart speakers.

(2) After identifying the various devices in the target home, the attacker extracts packet-level features for each behavior of the devices based on relevant classification algorithms. They construct corresponding device behavior fingerprints and infer the behaviors of smart home devices, such as turning on/off smart lights and taking photos or videos with cameras. This allows them to obtain the behaviors of different types of devices and their corresponding timestamps within a certain period.

(3) The attacker performs logical analysis of device behaviors and, using machine learning or deep learning algorithms, establishes mapping relationships between device behaviors and user behaviors. This enables them to infer the underlying user behaviors behind the device behaviors. By continuously observing the target household over a period (e.g., a week or a month), the attacker can also deduce more in-depth privacy information about the user, such as the time when the user leaves the house or the periods when the camera is turned off.

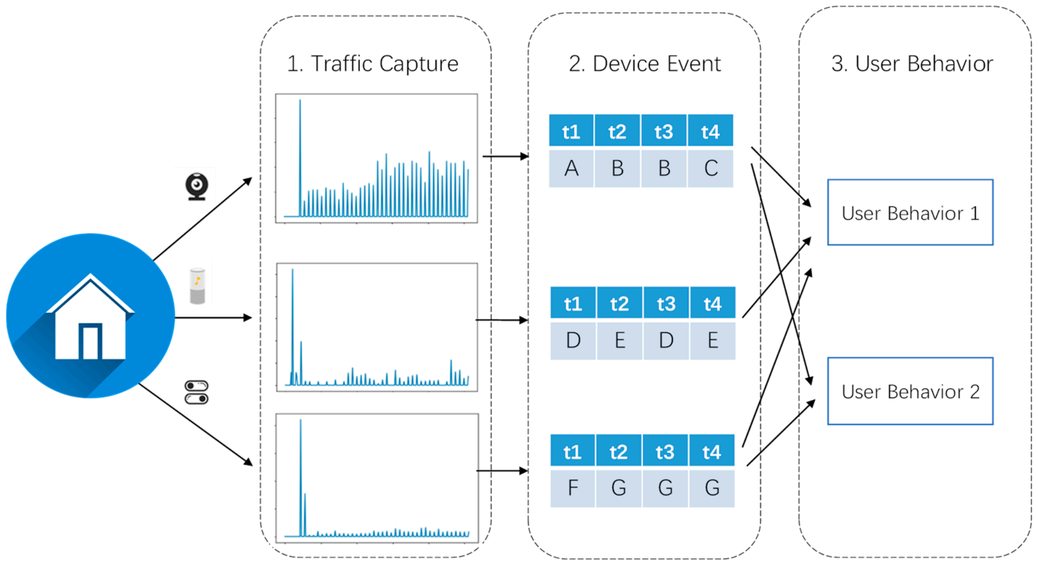

Figure 2. Attackers' goals of smart home traffic analysis.

3.2. Feature Selection

For traffic fingerprint training, there are various features can be extracted at the packet level, flow level, and behavior level. Packet-level features are those features which are extracted from each packet. These packet-level features including source and destination port, packet length, and packet payload length. A network flow is defined by its 5-tuples, which are source IP address, destination IP addresses, source port number, destination port number, and transport layer protocol [27]. A network flow usually represents one complete message exchange between a client and a server. Flow level features including flow length, flow ratio, flow payload length, flow duration. There are few features in each IoT device, which is independent of packet-level or flow-level features but depends on the application behavior of devices and are called behavioral level features. The behavior level features including NTP interval, DNS interval, transmission rate, DNS queries, cipher suites and sleep time.

For the features above, packet length, flow length, flow ratio, and transmission rate are often used for extracting device event fingerprint and infer privacy. Therefore, padding mechanism based on packet length should be an effective solution.

3.3. Goals and Challenges

Due to the relatively small scale of smart home device traffic data compared to network traffic and the high accuracy and real-time nature of packet-level features [28], most of the related research utilizes classification algorithms based on packet-level features for traffic classification by attackers. Therefore, this study designs attack models that infer user privacy information based on packet-level traffic fingerprints of smart home devices to demonstrate the feasibility and potential harm of smart home traffic classification attacks. These attack models are used to evaluate the effectiveness of the proposed SHTObfuscator defense method. It should be noted that SHTObfuscator also alters features such as transmission speed and data flow size, thereby providing a certain level of defense against attack models that rely on data flow-level features as well.

Our goals are as follows:

(1) Injecting traffic fingerprints of different device behaviors into the smart home traffic environment as obfuscation traffic, constructing a flow behavior trajectory for a virtual user, which can effectively reduce the effectiveness of user privacy inference models.

(2) The behavior logic of the virtual user is self-consistent and does not conflict with real user behavior, making it difficult for attackers to uncover the virtual user through

logical inference. Additionally, the injection of obfuscation traffic does not interfere with the normal operation of smart home devices and the daily lives of real users.

(3) The privacy protection level can be adjusted to control the number of obfuscated packets injected, achieving a balance between privacy protection and bandwidth overhead.

4. Design

4.1. Overall Design

The main idea of the proposed smart home traffic obfuscation method, SHTObfuscator is as follows:

As mentioned earlier, attackers can use traffic fingerprinting techniques to launch traffic classification attacks on smart home. Similarly, injecting obfuscation traffic based on device traffic fingerprints into real smart home traffic can reduce the classification effectiveness of traffic classification models. This makes it difficult for attackers to distinguish which traffic is generated by the devices, thus hindering their ability to infer the real user's behavior privacy, and ensuring user privacy to a large extent.

Meantime, previous research has shown that if the injected obfuscation traffic only involves a single device, or if the virtual device behavior lacks logical consistency or conflicts with the real user's behavior, attackers can use causal analysis or logic integrity checks to discern the virtual device behavior corresponding to the obfuscation traffic. Once attackers identify the obfuscation traffic, the defense mechanism of traffic obfuscation becomes ineffective. Therefore, to achieve better defense, simply simulating device behavior and injecting fake traffic from the device level is insufficient. Since a smart home user's behavior may involve interactions among multiple devices, simulating behavior at the user level is more effective in defense and deception.

Based on the above analysis, before generating obfuscation traffic, SHTObfuscator follows a series of steps to construct a logically consistent sequence of virtual user behaviors. It generates corresponding virtual device behavior sequences based on the mapping relationship between the target user's behavior and device behavior. Finally, using the fingerprint data saved in the smart home device traffic fingerprint library, obfuscation traffic is generated using a traffic generation tool and injected into the gateway traffic of the target smart home. This creates a seemingly realistic user behavior trajectory within the smart home traffic data, making it difficult for attackers to distinguish between inferred behaviors of real users and virtual users, thus ensuring user privacy.

As the number of virtual devices and behaviors increases, attention must be given to the increase in bandwidth overhead. To address this concern, this paper proposes a "tunable" privacy protection level for the traffic obfuscation scheme, allowing users to dynamically balance privacy protection and cost overhead based on their home's bandwidth conditions and requirements.

The traffic obfuscation process of SHTObfuscator consists of three functional stages: traffic fingerprint extraction, virtual user generation, and obfuscation traffic injection. The processes and detailed steps of each functional stage are depicted in Figure 3.

4.2. Event Fingerprint Extraction

Obviously, an adversary could easily use traffic rate changes to infer user activities. There are many works trying to classify different smart home devices based on their traffic characteristics. Features in different resolutions, including packet level, flow level and behavior level are extracted as fingerprints and then fed into machine learning models to identify the type of the device which generates the traffic [29].

For example, Trimananda et al. [5] proposed a smart home device activity classification method. By extracting the size and direction characteristics of smart home traffic data packets, a unique fingerprint was established for each activity of each device, and the activities (such as light bulb on/off) were classified by detecting the fingerprint. They identified the traffic flows that occurred immediately after each event and observed that certain pairs of packets with specific lengths and directions followed each ON/OFF event:

the same pairs consistently showed up for all events of the same type (e.g., ON), but were slightly different across event types (ON vs. OFF). The pairs were comprised of a request packet in one direction, and a reply packet in the opposite direction. Intuitively, this makes sense: if the smart home device changes state, this information needs to be sent to (request), and acknowledged by (reply), the cloud server to enable devices that are not connected to the home network to query the smart home device's current state.

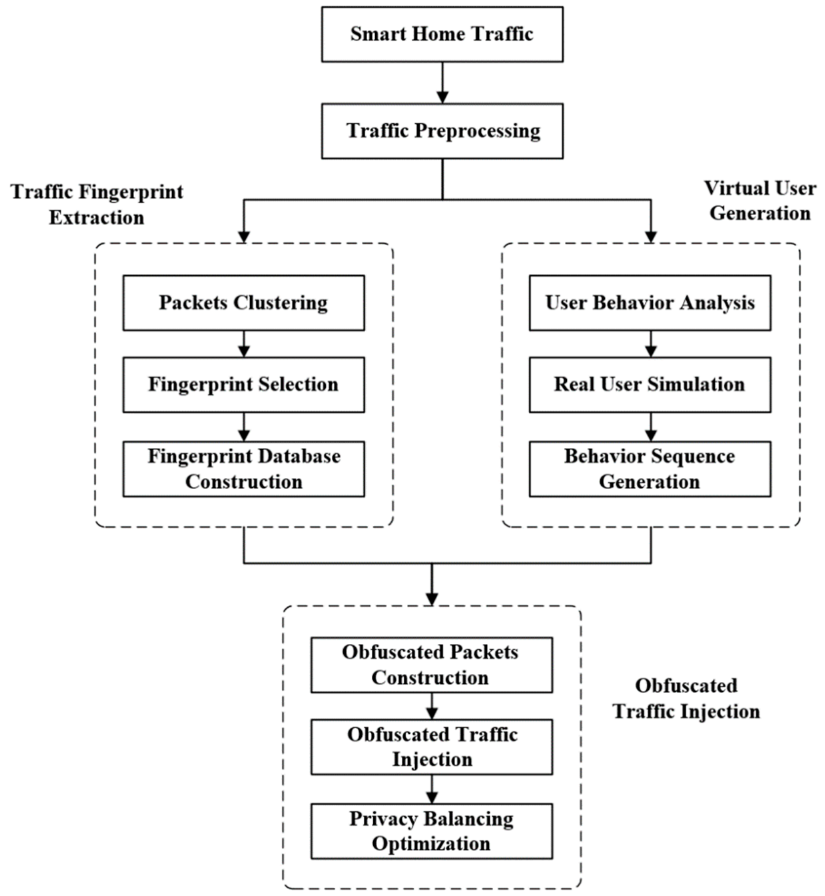
Smart Home Traffic
Traffic Preprocessing
Traffic Fingerprint Extraction
Virtual User Generation
Packets Clustering
Fingerprint Selection
Fingerprint Database Construction
User Behavior Analysis
Real User Simulation
Behavior Sequence Generation
Obfuscated Packets Construction
Obfuscated Traffic Injection
Privacy Balancing Optimization
Obfuscated Traffic Injection

Figure 3. Processes of our proposal.

With the approach of [5], we extracted the stream on/off fingerprints of Xiaomi smart camera, as shown in Figure 4, we observed an exchange of 2 TLS Application Data packets between the plug and an Internet host where the packet lengths were 115, 299, 364 and 1061 when the stream was toggled ON, but 445 and 442 for OFF. This preliminary analysis indicates that each type of event is uniquely identified by the exchange of pairs (or longer sequences) of packets of specific lengths.

Once the candidate fingerprints are selected, their usability as device behavior fingerprints is verified using a detection algorithm. If the maximum number of device behaviors detected using the device behavior fingerprint is the same (or similar) as the actual number of device behaviors, and the timestamps of the detected device behaviors match the timestamps of the events recorded during the training period, the device behavior fingerprint is finally determined to be a valid device behavior fingerprint and stored in the device behavior fingerprint file.

Device behavior fingerprints can be generated for multiple devices and their different behaviors, forming a dynamically maintained device behavior fingerprint library. The device behavior fingerprint library includes the device types and traffic fingerprints of various behaviors. Since device behavior fingerprints may change with changes in manufacturer software or protocol versions, regular updates and maintenance are required.

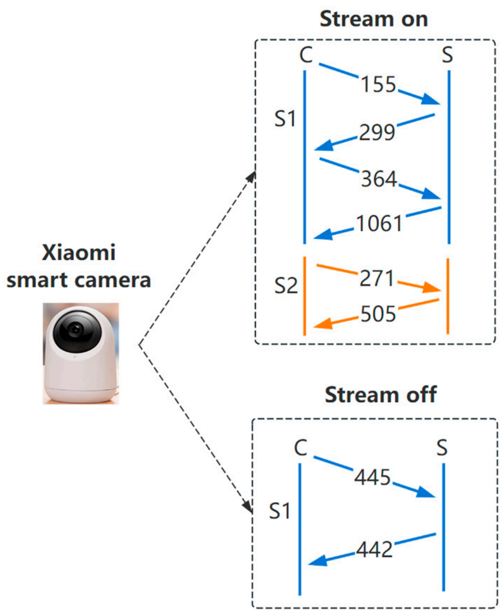

Figure 4. Stream on/off event fingerprints of Xiaomi smart camera.

4.3. Virtual User Generation

4.3.1. User Behavior Analysis

The design objective of the virtual user is to make the attacker perceive them as a real person living in the home based on the traffic characteristics. Therefore, the behavior associations or habits of the virtual user should closely resemble those of the target home's real user. At the same time, the behavior of the virtual user should be sufficiently different from that of the real user to prevent the attacker from distinguishing between the two. The behavior of the virtual user should be logically consistent and not conflict with the behavior of the real user.

The effectiveness of the virtual user concept lies in how to construct a reasonable sequence of home behaviors for the virtual user. Before doing so, it is important to clarify which smart home devices and their corresponding behavior activities are associated with each user behavior pattern. This may vary significantly for different users due to their individual behavioral habits. Therefore, it is necessary to establish a mapping relationship between user behaviors and device behaviors in the target home, ensuring that each user behavior includes a sequence of behaviors from one or more devices. This device behavior sequence not only includes different types of devices but also captures the sequential relationships between behaviors of different devices. Therefore, the behavior B_a of a real user a in a smart home can be represented by the Formula (1).

$$B_a = \{< D_1,\ Event_1,\ Time_1 >,\ \ldots\ ,\ < D_i,\ Event_i,\ Time_i >\} \tag{1}$$

The D_i represents the smart home devices present in the household, $Event_i$ represents the associated device behavior, and $Time_i$ represents the occurrence time of the device behavior. These three parameters can be used to describe the behavior of a smart home user. In order to model the behavior of smart home users and virtual users in future research, it is important to have a clear understanding of their actions and patterns.

Since the captured traffic data are continuous, it is necessary to segment the device behavior streams in the traffic files to ensure that each segmented device behavior stream accurately represents a user behavior and establish the mapping relationship. Then, based on these segments, the device-level features of user behavior are extracted, allowing for the selection of activities to be built for the virtual user and the generation of obfuscated traffic based on identifiers. Therefore, the first step is to reasonably segment the device behavior streams in the traffic data.

(1) Device Behavior Segmentation

Due to the variety and granularity of current smart home devices, directly identifying and classifying device behaviors from raw device traffic data can be complex. Therefore, it is possible to first divide the scenarios of smart home users' daily life patterns. For example, the user's daily pattern can be divided into scenarios such as waking up, cooking, leaving home, coming back home, and sleeping based on time relationships. The main types of devices involved in each pattern can be predefined. For instance, the "coming back home" pattern often involves changes in the status of devices such as smart locks, smart lights, and smart cameras. Then, based on the captured traffic transmission time, the current user scenario X_i can be preliminarily determined, and the approximate range of devices can be defined. The corresponding device traffic fingerprints can be extracted from the fingerprint database, and device behaviors can be identified and classified by matching the traffic fingerprints. This approach can greatly improve the efficiency of segmentation.

Specifically, device traffic can be divided into $B_a = \{b_1, \ldots, b_i\}$, where b_i represents a user behavior, denoted as $b_i = \{e_1, \ldots, e_j\}$, and e_j is one device behavior within it. The segmentation is based on three factors: time intervals between traffic fingerprints, proximity between traffic fingerprints, and the frequency of occurrence of traffic fingerprints [30], as described below:

(i) Time intervals between traffic fingerprints: By observing the time intervals between traffic fingerprints, device behaviors can be divided into different time periods or stages. For example, device behaviors with short time intervals can be grouped together, indicating that the user performed multiple consecutive device operations within a short period. Device behaviors with long time intervals can be grouped separately, indicating that the user performed the next device operation after a longer time.

(ii) Proximity between traffic fingerprints: This factor measures the correlation and continuity between device behaviors. If there is a strong proximity between the user's device behaviors, they can be grouped together. For example, if the user sequentially turns on the TV, sound system, and lights, indicating a high proximity between these devices, they can be grouped together.

(iii) Frequency of occurrence of traffic fingerprints: By observing the frequency of occurrence of traffic fingerprints, device behaviors can be categorized. For example, if a user frequently turns on a particular device, those frequently occurring device behaviors can be grouped together, while infrequently occurring device behaviors can be grouped separately.

By considering these three factors, device behavior streams can be effectively segmented and classified.

(2) User Behavior Feature Extraction

Based on segmenting and classifying device behavior streams, we can deduce which device behaviors belong to the same user behavior and establish an initial mapping relationship. However, to make the mapping relationship more comprehensive and complete, further exploration of user behavior features is needed.

Firstly, through observation and analysis, it is evident that smart home user behavior data exhibit symmetry. This symmetry refers to the regularity and consistency in user device operations. Specifically, the symmetry behavior features of smart home users can be manifested in the following aspects:

(i) Symmetry in turning on and off: The symmetry behavior feature of smart home users can be reflected in the activation and deactivation of devices. For example, if a user turns on the lights in the living room at a certain time, according to the symmetry feature, the user is likely to turn off the lights at some future time to maintain the symmetry of device behavior.

(ii) Symmetry in temporal patterns: The symmetry behavior feature of smart home users can also be observed in the temporal patterns of device operations. User behaviors may exhibit symmetry within specific time periods during the day. For instance, if a user turns on the bedroom air conditioner at a specific time in the evening, according to the symmetry feature, the user may also turn off the air conditioner during the subsequent same time period.

(iii) Symmetry in device combinations: The symmetry behavior feature of smart home users can be reflected in device combinations. User behaviors may exhibit symmetry when simultaneously using multiple devices. For example, if a user turns on the television, they may also turn on the stereo system simultaneously, and then turn them off after a certain period to maintain the symmetry of device behavior.

Therefore, the behavior symmetry parameter M can be introduced to describe this user behavior feature.

Furthermore, based on relevant research [31], due to the significant randomness in smart home user behavior, the device behaviors within the same user behavior class can be divided into deterministic behaviors and non-deterministic behaviors. For example, consider a smart motion-sensing light installed at the entrance of a kitchen in a household. The user's cooking behavior would involve both deterministic and non-deterministic behaviors. When the motion sensor detects the user passing by, the light turns on. This device behavior is deterministic, occurring every time the user engages in cooking. Apart from deterministic behaviors, there are other behaviors as well. For instance, during the cooking process, the use of a microwave, refrigerator, or kettle is not always certain. These uncertain behaviors represent variation and randomness, while deterministic behaviors constitute inherent characteristics of user behavior. In such cases, the device behaviors associated with a specific user behavior can be represented as $db \cup nb$, where db represents deterministic behaviors and nb represents non-deterministic behaviors.

In summary, the mapping relationship BD between user behavior and device behavior can be described using the formula:

$$BD = ((db + M_{db}) \cup (nb + M_{nb})) \tag{2}$$

(3) User Behavior Connections Exploration

In real smart home environments, user behaviors often follow certain patterns known as behavioral habits. To effectively protect the privacy of target households' user behaviors, this study creates deceptive virtual users whose behaviors differ from real users. To achieve this, it is necessary to uncover the regularities and patterns of real users' behavioral habits. This study considers two main types of associations in the target household's behaviors: the association between user behaviors and time, and the association between user behaviors themselves.

The association between behavior and time represents the relationship between a behavior and the time it occurs, such as the user's habit of performing a specific action at a certain time of the day. Based on relevant studies [32], the probability of a behavior occurring at a specific time can be calculated by analyzing the behavioral records from the previous three days or a week, using a time unit of two hours. This probability, denoted as P_1, serves as a parameter for capturing the temporal relationships when constructing virtual user behaviors.

The association between user behaviors represents the occurrence of other behaviors when a particular behavior takes place, capturing the temporal order of behaviors based on real users' habits. The probability of two behaviors occurring together can be obtained through statistical analysis of the associations between user behaviors. This probability, denoted as P_2, serves as a parameter for capturing the behavioral relationships when constructing virtual user behaviors.

Based on the above probability parameters, a preliminary model can be developed to represent the behavioral habits of real users in smart home environments. This enables the generation of virtual user behaviors that are more aligned with real users, thus enhancing the deceptive nature of the generated behaviors. However, it should be noted that this model is still based on probabilistic parameters derived from statistical analysis and cannot accurately predict the specific behaviors of each user. Therefore, it is important to consider other practical factors to improve the accuracy and realism of the generated virtual user behaviors.

4.3.2. Behavior Sequence Generation

(1) Construction of Virtual User Behaviors

Based on the previous three steps and after obtaining the required information from the target smart home, the construction of virtual user behaviors begins. This study adopts a top-down approach to build virtual users. Firstly, the behavior sequences of virtual users are constructed. Then, for each user behavior, the corresponding sequences of device behaviors are generated. Finally, traffic fingerprints are extracted from the fingerprint library corresponding to each device behavior, and obfuscated traffic is injected into the target home.

According to Equation (1), based on the behavior patterns B_a of real user a, the daily behavior of virtual user a, denoted as V_a, can be described as:

$$V_a = \{< V_{e0},\ Time_0 >,\ \ldots,\ < V_{ei},\ Time_i >\} \quad (3)$$

V_{ei} represents a virtual user behavior, and $Time_i = < T_{begin},\ T_{end} >$ represents its time information, including the start time T_{begin} and end time T_{end}. For example, the behavior pattern "having breakfast from 7 a.m. to 8 a.m. and cooking from 11 a.m. to 12 p.m." is part of the user behavior pattern. To avoid suspicion from attackers, it is necessary to generate dynamic behaviors for virtual users each day. Based on the real users' behavioral habits learned in the target household, a function F is used to determine the virtual user's behavior V_{ei} and its occurrence time $Time_i$:

$$< V_{ei},\ Time_i >= F(X_i,\ P_1,\ P_2, Past_i,\ M_i) \quad (4)$$

X_i is the user's current context at time t_i, P_1 and P_2 are the probabilities of a certain user behavior occurring, obtained from the analysis of real user behavior habits in terms of time and behavioral associations, respectively. These probabilities can be derived from statistical analysis of the learned real user behavior patterns in the target home. $Past_i$ represents the behaviors that the virtual user has already performed at a specific time, and M_i is the result of the symmetry analysis of the real user's already occurred behaviors. By considering multiple factors, it is possible to simulate the real user's behavior and ensure the logical integrity of the virtual user's own behavior.

If there is a high similarity between real and virtual users, attackers may still be able to infer the real user's behavior [33]. Therefore, the behavior patterns of virtual users should be different from those of real users. Planning the occurrence time of virtual user behaviors ensures that real and virtual users are engaged in different activities. Based on the behavioral associations and temporal relationships described earlier, the next behavior of the virtual user is determined based on the previous behavior and its duration. Thus, the assigned behaviors to virtual users are logically reasonable.

Furthermore, to strike a balance between smart home privacy protection and bandwidth consumption, a privacy protection level f can be introduced. Different privacy protection levels correspond to different quantities of generated virtual user behaviors. As the privacy protection level f decreases, the number of generated virtual user behaviors decreases accordingly. This is because under high privacy protection levels, the generated virtual user behaviors should closely resemble those of real users to ensure effective obfuscation of real user traffic. However, in situations where network resources are scarce, an excessive number of virtual user behaviors can have a negative impact on network performance. Therefore, it is necessary to control the quantity of virtual user behaviors. Based on the analysis above, the daily behavior V_a of a virtual user can be generated using the following Equation (5):

$$V_a = D\ (F,\ f,\ < V_{e0},\ Time_0 >) \quad (5)$$

In the above statement, F is the function used to generate virtual user behaviors and their corresponding time. f is the adjustable privacy protection level, and V_{e0} represents the first behavior of the day, typically waking up. For a virtual user, the time of their waking up behavior, $Time_0$ can be randomly generated within a certain range. The algorithm for function D is shown in Algorithm 1. Throughout the day, based on the generated initial behaviors of the virtual user and referring to the probabilities of real user behavior habits and the already occurred user behaviors, the virtual user's behaviors are constructed step by step in chronological order.

Algorithm 1. Virtual User Behavior Generation

```
INPUT: Initial behavior of virtual user $<Ve_0, Time_0>$
Real user behavior probabilities $P_1, P_2$
Current context mode $X_i$, symmetry test parameter $M_i$
Confusion level parameter $f$
OUTPUT: Virtual user's daily behavior pattern
        $Va = <Ve_0, Time_0>, \ldots, <Ve_n, Time_n>$
DATA: SQLite Fingerprint Database $DB$
1.  Initialize $C(f)$ as the threshold for the number of virtual user behaviors under the
    predetermined $f$ level
2.  $count = 1$
3.  $Va = <Ve_0, Time_0>$
4.  for $Hour_i \in [0,23]$ do
5.  if $C(f)/24 >= count$ then
6.  $Past_i = <Ve_0, Time_0>, \ldots, <Ve_i, Time_{i-1}>$
7.  $<Ve_i, Time_i> = F(X_i, P_1, P_2, Past_i, M_i)$
8.  add $<Ve_i, Time_i>$ to $V_a$
9.  $count = count + 1$
10. end
11. end
```

(2) Device Event Sequence Generation

After constructing the virtual user's behavior sequence for a day, the next step is to generate the corresponding device behavior sequence based on the established mapping relationship. One key challenge is to ensure the contextual consistency and logical integrity of the device behavior sequence. Device behaviors are not only influenced by the current behavior but also by previous behaviors. For example, if the virtual user has previously turned on the lights in the living room, they should turn off the lights when leaving the room to maintain the coherence and consistency of device behaviors. Therefore, a symmetry test is also required for device behaviors.

The symmetry test for device behaviors can effectively reduce the occurrence of two unfavorable situations for privacy protection:

(i) Continuous virtual device behaviors that may unintentionally leak user privacy. For example, the sequence of three device behaviors: (real user) turn on the lights $\rightarrow$ (virtual user) turn off the lights $\rightarrow$ (virtual user) turn on the lights. In this case, an attacker can still determine the real state of the lights. To address this, when generating virtual user device behaviors, we can check if the previous behavior was a real "turn on" behavior and ensure that subsequent behaviors logically match it.

(ii) Logical conflicts between consecutive virtual device behaviors that render the confusion ineffective. For example, the sequence of three device behaviors: (virtual user) turn off the lights $\rightarrow$ (virtual user) turn off the lights $\rightarrow$ (virtual user) turn off the lights. When the same behavior occurs consecutively, an attacker may realize the presence of a virtual user, which could compromise the privacy protection. To mitigate this, when the virtual user attempts to turn off a device that is already off, a symmetry test can be performed to validate and correct the logical behavior.

Additionally, continuous monitoring of the target smart home is necessary. This includes monitoring the real home environment to ensure the virtual user's behavior aligns with the real user's behavior and promptly detecting any anomalous behaviors. In the same smart home environment, device behaviors should not create logical conflicts between home users (real and virtual), so the virtual user's behavior model needs to be updated accordingly.

The algorithm for generating the virtual user's device behavior sequence is presented in Algorithm 2. It involves symmetry behavior testing and deterministic behavior analysis based on the obtained user behavior sequence. The occurrence times of device behaviors are then sorted according to Equation (4), resulting in the final corresponding virtual device behavior sequence.

Algorithm 2. Device Behavior Sequence Generation

```
INPUT: Virtual user behavior sequence $V_a$
Symmetry test parameter $M_i$
Association between user behavior and device behavior $BD$
Confusion level parameter $f$
OUTPUT: Device behavior sequence $E_a = \{de_1, de_2, \ldots, de_n\}$
1.  Decompose $V_a$ into user behavior and time sequence
2.  $<Ve_0, Time_0>, \ldots, <Ve_m, Time_m>$.
3.  $count = 1$
4.  while $count \leq m$ do
5.  $i = count$
6.  $de_i = F(d_0, t_0, BD)$
7.  if $M_i$
8.  add $de_i$ to $Ea$
9.  else
10. continue
11. $count = count + 1$
12. end
```

4.4. Obfuscated Traffic Injection

After generating the behavior of virtual users and the corresponding device behavior sequences, the corresponding device behavior traffic fingerprints will be extracted from the fingerprint library and injected into the smart home outbound traffic through the gateway.

The construction process of obfuscating data packets is as follows: based on the captured original network traffic, the packet size and direction are determined according to the traffic fingerprint. By using MAC addresses that are similar or identical to real devices, attackers can be misled in the first step of identifying all device types within a home, thus affecting the subsequent stages of behavior identification. Additionally, virtual devices can be constructed using MAC addresses that are identical to real devices, effectively obfuscating the device behavior of a specific device and making it difficult for attackers to distinguish whether the identified device behavior comes from a real device. The obfuscated data packets use the same destination/source IP, port numbers, etc., as the real device traffic. The payload can be filled using relevant packet construction tools and dynamically adjusted.

Many smart home devices have bidirectional traffic interactions, such as smart voice assistants, which frequently exchange data with device cloud services during operation. The behavior traffic fingerprints of such devices are mostly bidirectional. To simulate this bidirectional interaction at the gateway exit, traffic obfuscation units can be deployed on both the local gateway and remote servers. When the gateway transmits traffic outward, the remote server masquerades as a role in the smart home device cloud service, responding to the "traffic demands" of the local virtual devices by sending obfuscated traffic to the gateway server. Within the limits of cost, the number of remote servers can be increased, establishing a many-to-one mapping relationship with the smart home gateway. This way, if one remote server is unable to work due to an attack or other reasons, other servers can still provide obfuscation support, ensuring the reliability and stability of privacy protection methods. By injecting bidirectional traffic between the gateway and cloud services, the obfuscated traffic becomes more covert, making it difficult for attackers to differentiate between real and virtual device traffic and thus making it challenging for attackers to target smart homes.

Based on the above, traffic obfuscation methods often incur certain network bandwidth overhead, which may increase the latency of smart home devices and affect user experience. For sensitive devices such as smart speakers that require high network performance, excessive network latency can also impact the normal functionality of the device. Therefore, to strike a better balance between privacy protection in smart homes and bandwidth overhead, SHTObfuscator provides three different levels of obfuscation intensity: Level I, Level II, and Level III. Under different obfuscation levels, the density of virtual user behaviors will correspondingly increase or decrease, resulting in flexible changes in the bandwidth overhead caused by obfuscated traffic. Smart home users can dynamically adjust the obfuscation intensity based on the current network conditions at home to achieve the best outcome.

According to related research [34], real smart home users generate an average of 30 behaviors per day. Therefore, for Level I obfuscation, approximately 15 user behaviors can be generated for each virtual user per day. For Level II obfuscation, approximately 30 user behaviors can be generated per virtual user per day. For Level III obfuscation, approximately 45 user behaviors can be generated per virtual user per day. By changing the obfuscation intensity to modify the number of user behaviors, the required size of obfuscated traffic can be adjusted, achieving a balance between privacy protection and bandwidth consumption optimization.

Based on observations of laboratory smart home devices, it has been found that the effectiveness of the attack model varies between day and night, with less privacy information exposed during the night. This is because fewer smart home devices are active during the night when users are asleep. Therefore, it is possible to reduce the obfuscation intensity during nighttime to save costs. Since there is a significant difference in the number of user behaviors between daytime and nighttime for real users, and the purpose of designing virtual users is to make attackers perceive them as real individuals, it is unnecessary to obfuscate the traffic patterns of daytime and nighttime into the same pattern. This functionality can be implemented in the subsequent system implementation to be automatically handled by the system itself without requiring manual switching by users. Of course, this should be done under the premise of confirming that the user is not staying up late or has already fallen asleep.

Meanwhile, in the subsequent system implementation, an adaptive obfuscation method can be designed to allow the system to determine the current time and automatically adjust the obfuscation intensity based on different time periods, thereby ensuring the balance between the security of privacy information and cost-effectiveness. For example, during nighttime periods, the system can choose to appropriately reduce the obfuscation intensity to reduce resource consumption, while during daytime periods, the system can use stronger obfuscation intensity to ensure the security of privacy information.

This adaptive obfuscation method can be implemented in the system through programming languages. The current time can be obtained using modules such as "datetime" in Python, and then the obfuscation intensity can be adjusted based on conditional statements according to the time period. At the same time, the actual usage patterns of users, such as their sleep time and habits, should be considered during the implementation process to avoid excessive or insufficient obfuscation when users need to use smart home devices.

In conclusion, the adaptive obfuscation method can balance the security of privacy information while saving network bandwidth resources and improving the cost-effectiveness of the system. During implementation, the user's usage habits and time factors should be considered to make the system more intelligent and flexible.

5. Experimental Evaluation

5.1. Experimental Setup

The padding process is application independent and can be performed at the transport, network, and link layers. The padding mechanism can be implemented in software, acting as a middlebox on devices, such as a home router and gateway.

The experimental environment topology, as shown in Figure 5, consists of seven types of smart home devices such as smart cameras and smart speakers, as well as non-smart home devices such as smartphones and computers. These devices are connected to the smart home gateway via WiFi and interact with a cloud server, forming a smart home living scenario. The smart home gateway is deployed with the SHTProtector traffic privacy protection system, and the remote server supports bidirectional injection of obfuscated traffic.

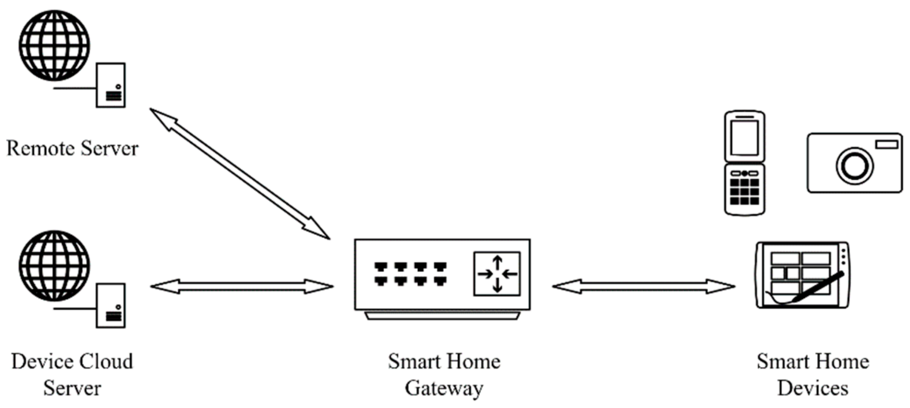

Figure 5. Experimental environment.

The hardware and software information of the smart home gateway based on OpenWrt is provided in Table 1.

Table 1. Hardware and Software Information.

Component	Specifications
CPU	i5-8400
RAM	8 GB
OpenWrt	19.07
DNSmasq	2.8.5
Hostapd	v2.10-devel

The real environment of the experiments is shown in Figure 6, including the smart home gateway and smart home devices (including cameras, smart speakers, and smart sensors).

The dataset used to generate traffic fingerprints consists of device traffic captured in the real laboratory environment. The laboratory environment dataset includes behavioral traffic and idle traffic from 10 devices. The device types include 7 smart home devices and 3 non-smart home devices, as shown in Table 2. The collection time for idle traffic is 10 min, while the collection of behavioral traffic is manually triggered based on the functional behaviors of each device. During the collection process, only the target smart home devices and non-smart home devices are in an active state. Each device behavior is triggered 30 times, with each trigger lasting 10 s, and the time of each behavior trigger is recorded.

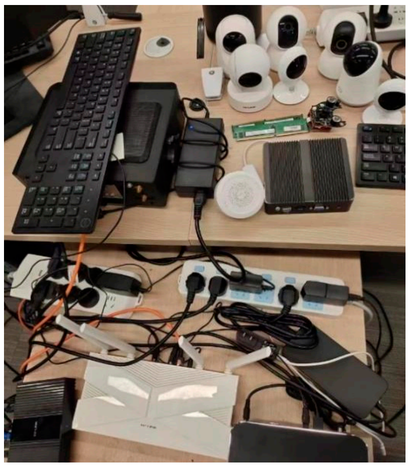

Figure 6. Real environment of experiments.

Table 2. Laboratory Device Information.

Manufacturer	**Name**	**Function**	**Device Behavior**
Lenovo	Camera	Fingerprinting	View monitoring
Xiaomi	Smart Pan-Tilt Camera	Fingerprinting	Recording
Hikvision	Ezviz Camera	Fingerprinting	Recording
Baidu	Xiaodu Speaker	Fingerprinting	Conversation
Aqara	Smart Light	Fingerprinting	On/Off
Aqara	Motion Sensor	Fingerprinting	Motion detection
Aqara	Smart Switch	Fingerprinting	On/Off
Huawei	Mate20 Smartphone	Background	Video browsing
Lenovo	IdeaPad 16 Laptop	Background	File downloading
Lenovo	ThinkPad Laptop	Background	Standby

To evaluate the functionalities of the SHTProtector smart home privacy protection system, the following experiments were designed to validate the effectiveness of various modules, including the deceptive nature of virtual traffic, the effectiveness of traffic obfuscation methods, and the efficacy of privacy balancing optimization.

5.1.1. Traffic Obfuscation Effectiveness Experiment

This experiment consists of two parts: device behavior recognition experiment and user behavior inference experiment.

(i) Device Behavior Recognition Experiment: The objective is to verify the deceptive nature of virtual traffic. Assume that an attacker extracts device behavior fingerprints from the dataset and trains a random forest classification model, this model is used to detect whether injected packets can be identified by the attacker as the expected device behavior and whether fake packets can be distinguished from real packets.

(ii) User Behavior Inference Experiment: The purpose is to validate the effectiveness of the obfuscation methods. Using a commonly used Hidden Markov Model (HMM) in the smart home domain, user behavior inference is performed to evaluate the obfuscation effect of the proposed traffic obfuscation method on the user behavior inference model.

5.1.2. Traffic Obfuscation Overhead Experiment

The goal of this experiment is to test the obfuscation effect and bandwidth overhead under different defense levels, in order to verify the balance between privacy protection strength and bandwidth overhead. The experiment evaluates the obfuscation effect and household traffic size under different defense levels and includes a control group for comparison, aiming to validate the effectiveness of privacy balancing optimization.

5.2. Efficiency of Obfuscation

Assume that attackers extract device behavior fingerprints from a dataset, they trained a random forest classification model. This model is used to detect whether injected packets can be recognized by the attacker as the intended device behavior and whether fake packets can be distinguished from real ones.

The results of the device behavior inference experiment are presented in Table 3. The recall rate of device behavior inference represents the ratio of successfully inferred behaviors to the total number of behaviors. The precision of device behavior inference represents the ratio of correctly inferred events to the total number of inferred events. The F1 score is the harmonic mean of both metrics. From the experimental results, it can be observed that the recall rate and F1 score of virtual packets used for confusion are very similar to those of real packets. This indicates that attackers tend to identify the behavior of virtual users as real device behavior, thus validating the high deceptive nature of the generated confusing traffic based on the proposed method in this paper. It effectively confuses attackers in real-world scenarios.

Table 3. Device Behavior Recognition Results.

Device Behavior		**Real**	**Virtual**
Device	**Behavior**	**F1 Score (%)**	**F1 Score (%)**
Xiaomi Camera	Record	91.7	91.1
Aqara Switch	Turn On	98.0	98.3
Xiaodu Speaker	Talk	91.5	93.0
Smart Light	Turn On	97.4	96.3
Motion Sensor	Sensor	96.6	96.5

After inferring device behavior, attackers need to construct a new classification model to infer user behavior. Previous research [35] has shown that Hidden Markov Models (HMMs) perform well in inferring user behavior in smart home environments. Based on observations and usage of smart home devices in a laboratory environment, the following analysis is presented:

In a smart home environment, user behavior typically involves multiple smart home devices. For example, the user's behavior of returning home may involve actions such as unlocking the door, turning on lights, adjusting the thermostat, and activating smart plugs. In such cases, attackers can use HMMs to infer user behavior from the states of smart home devices. The fundamental assumption of HMMs is that the hidden state forms a Markov chain, meaning that the current hidden state only depends on the previous hidden state and is independent of previous states. Therefore, when attackers observe a sequence of states from smart home devices, they can use HMMs to predict the sequence of hidden states (i.e., user behavior) and infer the user's privacy information.

Since the effectiveness of classifying device behavior attacks has been experimentally validated in the previous step, it is possible to identify the device behavior sequence within a specific time period in the home using a classification model. Therefore, a training dataset can be created, which includes sequences of device behavior and their corresponding user behaviors. Attackers can use this dataset to train an HMM and utilize the HMM to infer user behavior.

Table 4 shows the user behaviors and the underlying device behaviors involved. As shown in Table 5, the experimental results of user behavior inference before and after traffic obfuscation demonstrate that the proposed traffic obfuscation method has a good effect on confusing the classification model for user behavior inference. It effectively reduces the inference accuracy of the classification model for user behaviors such as leaving home, returning home, sleeping, waking up, and walking. The detailed experimental results can be found in the Appendix A.

Table 4. User and Device Behavior.

User Behavior	Relevant Device Behavior
Control Smart Light	Turn on, turn off, change color, adjust brightness
Control Smart Plug	Turn on, turn off
Talk to Smart Speaker	Turn on and engage in conversation with smart speaker
Sleep	Activate camera, turn on motion sensor, turn off smart light
Wake Up	Turn off motion sensor, turn off camera, turn on speaker
Leave Home	Turn off motion sensor, turn on camera, turn off light, unlock/lock door
Back Home	Unlock/lock door, turn on light, turn on motion sensor, turn off camera

Table 5. User Behavior Recognition Results under Different Defense Levels.

	Original	Level I	Level II	Level III
Behavior	**F1 Score (%)**			
Leave Home	92.2	31.4	27.3	22.6
Return Home	91.4	29.1	23.1	21.4
Sleep	90.6	28.0	24.6	19.3
Wake Up	91.3	26.3	22.7	18.0
Walk	94.5	25.6	22.8	19.5

Based on the key steps of the proposed SHTObfuscator method, the traffic obfuscation process was simulated on the dataset. The obfuscated traffic was then tested using the HMM user behavior inference model mentioned earlier. In addition, packet padding and traffic shaping methods were included as a comparison. The detailed experimental results, shown in Figure A1, demonstrate that the three obfuscation levels of the SHTObfuscator method significantly reduce the F1 scores of the attack model for all seven user behaviors, indicating effective privacy protection.

Among the compared traffic obfuscation methods, the effectiveness of packet padding surpasses that of obfuscation levels I and II. This is likely due to the ability of packet padding to directly alter the fundamental feature of packet size, providing more direct obfuscation for simple user behaviors such as controlling a single device. However, packet padding requires high granularity and results in significant bandwidth consumption.

Under the highest obfuscation level III, SHTObfuscator outperforms packet padding in obfuscating complex user behaviors, and the obfuscation levels I and II also yield better results compared to traffic shaping. Through these experimental comparisons, it is confirmed that the SHTObfuscator method exhibits excellent privacy protection performance.

5.3. Overhead Evaluation

The traffic obfuscation overhead experiment primarily focuses on comparing the network bandwidth consumption of SHTProtector at different obfuscation levels and testing the functionality of the privacy balancing module. The experiment involves collecting and comparing the sizes of traffic generated by smart home devices and their corresponding obfuscation effects (measured by the F1 score of the obfuscated attack model) over the course of one week after deploying the gateway. The average values are calculated for each obfuscation level.

Table 6 presents the results of the privacy balancing optimization experiment. For the three different levels of traffic obfuscation, the SHTProtector privacy protection system incurs approximately 6.7–17.8% additional bandwidth consumption, resulting in a decrease in the F1 score of the attack model from around 95% to 29.4–20.6%, with an average of approximately 25%. Compared to the control group methods such as packet padding, traffic shaping, and fake traffic injection, SHTProtector achieves a good balance between obfuscation effectiveness and lower bandwidth consumption.

Table 6. Overhead Evaluation of Padding Strategy.

Method	Effectiveness (%)	Traffic	Overhead (%)
Original Traffic	-	310 MB	-
Obfuscation Level I	29.4	331 MB	6.7
Obfuscation Level II	25.5	346 MB	11.6
Obfuscation Level III	20.6	365 MB	17.8
Packet Padding	21.3	-	87.5
Traffic Shaping	33.2	-	29.0
Fake Traffic Injection	28.3	-	31.1

Compared to existing related research, our study has made significant improvements to the smart home traffic obfuscation method based on the concept of virtual users. We have adopted a bidirectional traffic injection approach, enabling the obfuscated traffic to more realistically simulate the interactions among smart home devices. By injecting obfuscated traffic into the network, we increase the difficulty for attackers to analyze and identify traffic patterns, thereby enhancing the effectiveness of privacy protection.

It is worth noting that we have carefully considered the issue of logical integrity during the process of injecting obfuscated traffic. The obfuscated traffic not only effectively hides the users' real behaviors but also ensures that it does not cause system anomalies or data loss. Through a well-designed bidirectional traffic injection strategy, we maintain the coherence and interpretability of the traffic, making the obfuscated traffic more realistic and less detectable by potential attackers.

In addition to improving the quality of obfuscated traffic, this study also addresses the problem of bandwidth consumption during the traffic obfuscation process. We have achieved adaptive control of traffic obfuscation, allowing users to choose appropriate levels of protection based on their specific needs and privacy requirements. This balanced approach between privacy protection and cost control provides smart home users with more personalized privacy protection solutions.

Overall, the proposed smart home traffic obfuscation method based on virtual users not only enhances privacy protection but also considers the feasibility and practicality of traffic obfuscation. It offers an effective and feasible solution for data security and privacy protection in smart home environments, providing strong support for the security and privacy of future smart home systems.

6. Conclusions and Prospect

In recent years, the smart home industry has experienced rapid growth. While smart home devices provide users with convenient and intelligent living experiences, they also reflect users' household behaviors in traffic data, which poses privacy risks. To address the issue of smart home traffic privacy protection, this paper proposes an enhanced smart home traffic obfuscation method called SHTObfuscator, and designs and implements a smart home traffic privacy protection system called SHTProtector. The main contributions of this paper are as follows:

(1) Based on the concept of virtual users, an improved traffic obfuscation method, SHTObfuscator, is proposed. This method injects traffic fingerprints of different device behaviors into the real traffic environment of smart homes as obfuscated traffic. It effectively prevents attackers from distinguishing the real device operation status and user behavior privacy in the home, thereby reducing the effectiveness of traffic classification attack models. It also provides different levels of protection to achieve a balance between privacy protection and bandwidth overhead.

(2) A virtual user behavior construction method based on logical integrity is proposed. This paper classifies the different behaviors of virtual users and their corresponding device behaviors and considers the logical relationship between virtual user behaviors and real user behaviors. The method ensures that the constructed virtual user behaviors have a high level of deception while performing traffic obfuscation.

(3) The design and testing of an adaptive smart home traffic privacy protection system. Based on the SHTObfuscator traffic obfuscation method, a smart home traffic privacy protection system, SHTProtector, is designed and implemented. It includes functions such as device identification, traffic fingerprint extraction, obfuscated traffic injection, and privacy balance optimization. Experiments are conducted in an experimental environment composed of smart home gateways and smart home devices to evaluate device identification monitoring, traffic fingerprint extraction, traffic obfuscation effectiveness, and traffic obfuscation overhead, thereby validating the effectiveness of the proposed methods.

Based on the research on smart home traffic obfuscation methods, further exploration is needed in the following areas:

(1) It is necessary to continue investigating the bandwidth consumption introduced by the virtual user based smart home traffic obfuscation method in real environments and find a better balance between privacy protection strength and bandwidth consumption.

(2) This paper primarily focuses on the case of a single virtual user. Further research is needed to explore obfuscation methods for constructing multiple virtual users.

(3) For smart home privacy protection systems, further research is needed on how to make the system more intelligent in adjusting traffic obfuscation modes to achieve more efficient utilization of computational resources.

Author Contributions: Conceptualization, S.Z., F.S. and Y.L.; Methodology, S.Z. and F.S.; Software, F.S.; Validation, F.S.; Formal analysis, F.S. and Z.Y.; Investigation, Y.L.; Resources, S.Z.; Data curation, F.S.; Writing—original draft, F.S.; Writing—review & editing, S.Z., Y.L., Z.Y. and X.L.; Visualization, F.S., Z.Y. and X.L.; Supervision, Y.L.; Project administration, S.Z., Y.L., Z.Y. and X.L.; Funding acquisition, S.Z. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by The Major Key Project of PCL (Grant No. PCL2022A03), Key-Area Research and Development Program of Guangdong Province (No. 2019B010137005).

Data Availability Statement: Owing to the policies and confidentiality agreements adhered to in our laboratory, the data presented in this study are available on request from the corresponding author.

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

Appendix A

The detailed experimental results are shown in Figure A1.

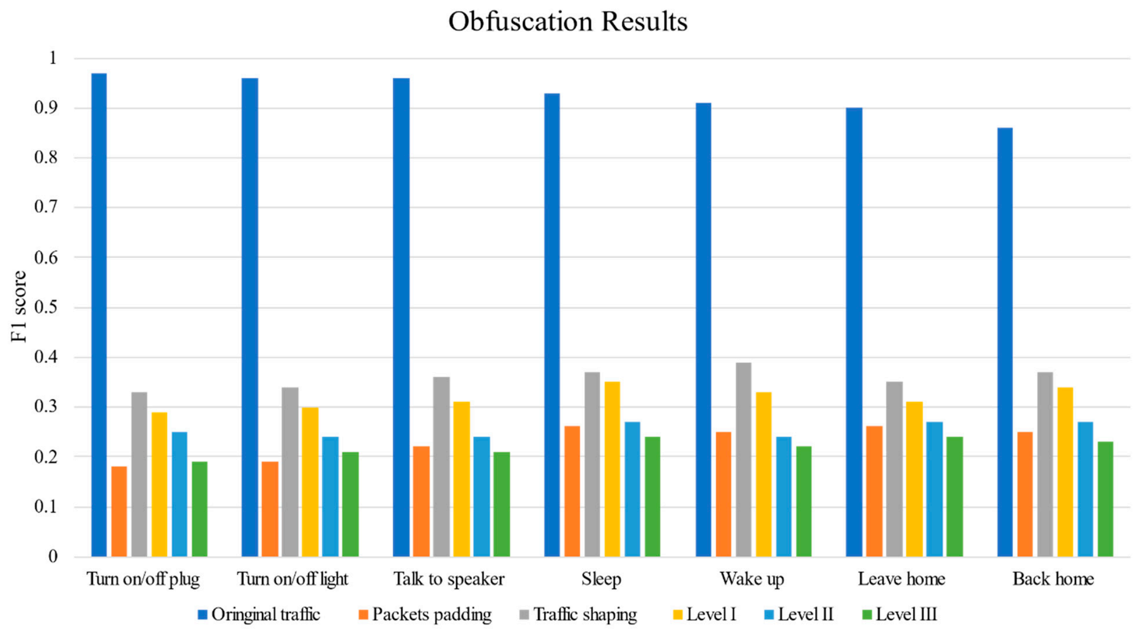

Figure A1. Detailed obfuscation results.

References

1. Cisco, U. *Cisco Annual Internet Report (2018–2023) White Paper*; Cisco: San Jose, CA, USA, 2020.
2. Acar, A.; Fereidooni, H.; Abera, T.; Sikder, A.K.; Miettinen, M.; Aksu, H.; Conti, M.; Sadeghi, A.R.; Uluagac, S. Peek-a-Boo: I See Your Smart Home Activities, Even Encrypted! In Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Linz, Austria, 8 July 2020; ACM: New York, NY, USA, 2020; pp. 207–218.
3. Salman, O.; Elhajj, I.H.; Kayssi, A.; Chehab, A. A Review on Machine Learning Based Approaches for Internet Traffic Classification. *Ann. Telecommun.* **2020**, *75*, 673–710.
4. Alshehri, A.; Granley, J.; Yue, C. Attacking and Protecting Tunneled Traffic of Smart Home Devices. In Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, 16–18 March 2020; pp. 259–270.
5. Trimananda, R.; Varmarken, J.; Markopoulou, A. Packet-Level Fingerprints for Smart Home Devices. In Proceedings of the 2020 Network and Distributed System Security Symposium, San Diego, CA, USA, 23–26 February 2020; pp. 1084–8045. [CrossRef]
6. Copos, B.; Levitt, K.; Bishop, M.; Rowe, J. Is Anybody Home? Inferring Activity from Smart Home Network Traffic. In Proceedings of the 2016 IEEE Security and Privacy Workshops, San Jose, CA, USA, 22–26 May 2016.
7. Dong, S.; Li, Z.; Tang, D.; Chen, J.; Sun, M.; Zhang, K. Your Smart Home Can't Keep a Secret: Towards Automated Fingerprinting of IoT Traffic with Neural Networks. In Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, Taipei, Taiwan, 5–9 October 2020.
8. Yao, Z.J.; Ge, J.G.; Zhang, X.D.; Zheng, H.B.; Zou, Z.; Sun, K.K.; Xu, Z.H. Research review on traffic obfuscation and its corresponding identification and tracking technologies. *J. Softw.* **2018**, *29*, 3205–3222. (In Chinese)
9. Nicolazzo, S.; Nocera, A.; Ursino, D.; Virgili, L. A privacy-preserving approach to prevent feature disclosure in an IoT scenario. *Future Gener. Comput. Syst.* **2020**, *105*, 502–519. [CrossRef]
10. Corradini, E.; Nicolazzo, S.; Nocera, A.; Ursino, D.; Virgili, L. A two-tier Blockchain framework to increase protection and autonomy of smart objects in the IoT. *Comput. Commun.* **2022**, *181*, 338–356. [CrossRef]
11. Pinheiro, A.J.; Bezerra, J.M.; Campelo, D.R. Packet Padding for Improving Privacy in Consumer IoT. In Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, Brazil, 25–28 July 2018.
12. Apthorpe, N.; Reisman, D.; Sundaresan, S.; Narayanan, A.; Feamster, N. Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic. *arXiv* **2017**, arXiv:1708.05044.
13. Xiong, S.; Sarwate, A.D.; Mandayam, N.B. Defending against packet-size side-channel attacks in IoT networks. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2027–2031.
14. Pinheiro, A.J.; de Araujo-Filho, P.F.; Bezerra, J.D.M.; Campelo, D.R. Adaptive packet padding approach for smart home networks: A tradeoff between privacy and performance. *IEEE Internet Things J.* **2020**, *8*, 3930–3938.
15. Wang, C.; Kennedy, S.; Li, H.; Hudson, K.; Atluri, G.; Wei, X.; Sun, W.; Wang, B. Fingerprinting encrypted voice traffic on smart speakers with deep learning. In Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Linz, Austria, 8–10 July 2020; pp. 254–265.
16. Prates, N.; Vergütz, A.; Macedo, R.T.; Santos, A.; Nogueira, M. A defense mechanism for timing-based side-channel attacks on IoT traffic. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
17. Ibitoye, O.; Matrawy, A.; Shafiq, M.O. A GAN-based Approach for Mitigating Inference Attacks in Smart Home Environment. *arXiv* **2020**, arXiv:2011.06725.
18. Ranieri, A.; Caputo, D.; Verderame, L.; Merlo, A.; Caviglione, L. Deep adversarial learning on google home devices. *arXiv* **2021**, arXiv:2102.13023.
19. Apthorpe, N.; Reisman, D.; Feamster, N. Closing the blinds: Four strategies for protecting smart home privacy from network observers. *arXiv* **2017**, arXiv:1705.06809.
20. Hafeez, I.; Antikainen, M.; Tarkoma, S. Protecting IoT-environments against traffic analysis attacks with traffic morphing. In Proceedings of the 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kyoto, Japan, 11–15 March 2019; pp. 196–201.
21. Zhu, Q.; Yang, C.; Zheng, Y.; Ma, J.; Li, H.; Zhang, J.; Shao, J. Smart home: Keeping privacy based on Air-Padding. *IET Inf. Secur.* **2021**, *15*, 156–168.
22. Xu, Z.; Khan, H.; Muresan, R. TMorph: A Traffic Morphing Framework to Test Network Defenses Against Adversarial Attacks. In Proceedings of the 2022 International Conference on Information Networking (ICOIN), Jeju-si, Republic of Korea, 12–15 January 2022; pp. 18–23.
23. Liu, X.; Zeng, Q.; Du, X.; Valluru, S.L.; Fu, C.; Fu, X.; Luo, B. Sniffmislead: Non-intrusive privacy protection against wireless packet sniffers in smart homes. In Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses, San Sebastian, Spain, 6–8 October 2021; pp. 33–47.
24. Liu, X.; Zeng, Q.; Du, X.; Valluru, S.L.; Fu, C.; Fu, X.; Luo, B. Privacyguard: Enhancing smart home user privacy. In Proceedings of the 20th International Conference on Information Processing in Sensor Networks (Co-Located with CPS-IoT Week 2021), Nashville, TN, USA, 18–21 May 2021; pp. 62–76.
25. Apthorpe, N.; Reisman, D.; Feamster, N. A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. *arXiv* **2017**, arXiv:1705.06805.

26. Datta, T.; Apthorpe, N.; Feamster, N. A developer-friendly library for smart home IoT privacy-preserving traffic obfuscation. In Proceedings of the 2018 Workshop on IoT Security and Privacy, Budapest, Hungary, 20 August 2018; pp. 43–48.
27. Dyer, K.P.; Coull, S.E.; Ristenpart, T.; Shrimpton, T. Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In Proceedings of the 2012 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–23 May 2012; pp. 332–346.
28. Asif, M.; Khan, T.A.; Taleb, N.; Said, R.A.; Siddiqui, S.Y.; Batool, G. A Proposed Architecture for Traffic Monitoring & Control System via LiFi Technology in Smart Homes. In Proceedings of the 2022 International Conference on Business Analytics for Technology and Security (ICBATS), Dubai, United Arab Emirates, 16–17 February 2022; pp. 1–3.
29. Apthorpe, N.; Huang, D.Y.; Reisman, D.; Narayanan, A.; Feamster, N. Keeping the smart home private with smart (er) iot traffic shaping. *Proc. Priv. Enhancing Technol.* **2019**, *2019*, 128–148.
30. Liu, J.; Zhang, C.; Fang, Y. Epic: A differential privacy framework to defend smart homes against internet traffic analysis. *IEEE Internet Things J.* **2018**, *5*, 1206–1217.
31. Jmila, H.; Blanc, G.; Shahid, M.R.; Lazrag, M. A survey of smart home iot device classification using machine learning-based network traffic analysis. *IEEE Access* **2022**, *10*, 97117–97141. [CrossRef]
32. Yoshigoe, K.; Dai, W.; Abramson, M.; Jacobs, A. Overcoming invasion of privacy in smart home environment with synthetic packet injection. In Proceedings of the 2015 TRON Symposium (TROnShOW), Tokyo, Japan, 9–11 December 2015; pp. 1–7.
33. Yoshigoe, K.; Dai, W.; Abramson, M.; Jacobs, A. Anomaly traffic detection and correlation in smart home automation IoT systems. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4053.
34. Uddin, M.; Nadeem, T.; Nukavarapu, S. Extreme SDN Framework for IoT and Mobile Applications Flexible Privacy at the Edge. In Proceedings of the 2019 IEEE International Conference on Pervasive Computing and Communications, Kyoto, Japan, 11–15 March 2019.
35. Hussain, A.M.; Oligeri, G.; Voigt, T. *The Dark (and Bright) Side of IoT: Attacks and Countermeasures for Identifying Smart Home Devices and Services*; Springer: Berlin/Heidelberg, Germany, 2021.