EncryptedIoT: A configurable and feature-rich E2EE policy for distributed IoT systems

Archit Agarwal
ara008@ucsd.edu
UC San Diego

Nagendra Jamadagni
njamadagni@ucsd.edu
UC San Diego

Rohit Pai
ropai@ucsd.edu
UC San Diego

Abstract

This paper presents an innovative end-to-end encrypted communication scheme designed to enhance security and privacy in Internet of Things (IoT) environments while maintaining usability. The proposed approach builds upon existing encryption protocols used by major IoT providers, specifically Amazon Ring, and introduces a novel concept of data stream keys (DSKs) and temporary data stream keys (TDSKs). These keys enable fine-grained control over data sharing without compromising the core benefits of end-to-end encryption. The research makes two primary contributions: First, it provides a detailed analysis of Ring's current encryption protocols, including both the default and opt-in end-to-end encryption schemes. Second, it proposes a modified protocol that allows users to retain exclusive control over their IoT device data while preserving access to a wide range of features typically disabled in existing end-to-end encryption implementations.

1 Introduction

IoT devices have experienced significant growth in relevance and popularity over the past decade, with smart home products becoming increasingly commonplace in households across the globe [10]. One of the biggest players in this industry is Amazon's Ring, which began as a video doorbell company, has expanded into a comprehensive smart home security ecosystem, selling over 1.7 million video doorbells in 2021 alone [5] and capturing a dominant market share. Another example is Google's Nest line of products, including smart thermostats, cameras, and speakers, has also gained traction, particularly after Google's acquisition of Nest in 2014 [1].

This raises significant security concerns from both data privacy, and usable security perspectives. In terms of data privacy, IoT devices collect extensive information from various sensors, such as video and audio feeds as well as data from environmental monitors like temperature sensors [3]. This data, which may be stored temporarily or permanently on provider-managed storage, must be protected so that neither the provider's operators nor potential attackers can access it.

The providers of these IoT devices are aware of the risks involved with storing such sensitive data. To mitigate these risks, they offer end-to-end encryption schemes that allow for user data to be fully encrypted when stored on their platforms [8]. However, these end-to-end encrypted offerings come at significant costs to device usability. Users are forced to forego all but a few bare bones features which makes enrolling in such schemes unattractive. [6]

2 Project Contributions

This paper has two main contributions. First, a detailed analysis of the end-to-end protocol used by Ring (The largest player by market share in this segment) along with a selection of notable attacks targeting these devices. Second, a proposal for a modified communication protocol enabling users to maintain exclusive control over their IoT device data, while still enjoying almost all features offered to them by IoT service providers.

The rest of this paper is organized as follows, Section 3 talks about two existing encryption schemes made available by Amazon Ring. The default security protocol and then the opt-in end-to-end encryption protocol. It also itemizes the list of features that are disabled for users who opt-in to the end-to-end encryption scheme. Section 4 details the proposed encryption scheme and demonstrates how this scheme allows users access to almost all features offered in the Ring ecosystem while maintaining the security guarantees of end-to-end encryption. Section 5 provides an evaluation of the proposed encryption scheme through Tamarin and demonstrates that it is verifiably safe to use. Section 6 provides final thoughts on the proposed as well as future work and discusses the contributions of each member.

3 Existing Encryption Protocols for Amazon Ring

In this section we first analyze the basic encryption features offered by Amazon's Ring range of products. Subsequently, we discuss the details of the opt-in end-to-end encryption feature offered by Amazon Ring.

3.1 Default Encryption Scheme of Ring IoT Devices

Ring to their credit have incorporated many standard security features and practices in their default encryption scheme. Ring has required two-factor authentication (2FA) since 2020 [9]. Any handshake or exchange of keys between the server and the user occurs using the TLS protocol. All live video and audio feeds that are being transmitted between the server to the client device are kept secure by the use of scrambled MPEG-TS packets. This was verified by capturing the network packets between the server and client device during an active session through the Wireshark tool and observing the collected data. Amazon also claims that any data from Ring devices stored on Amazon's servers is encrypted to safeguard them against leaks and attacks [6].

The data that is collected by Ring and stored on Amazon's servers includes information like audio and video feeds of a client's home. Amazon also stores a user's biometrics, address, and payment information. Although Amazon claims this highly sensitive information is stored in an encrypted form by Ring, there is no information provided by Amazon to back this claim. The encryption protocol as well as any keys used to encrypt this data on the server-side is not provided to the user.

Another set of concerns with the default encryption scheme is that this scheme is vulnerable to Man-In-The-Middle (MITM) attacks where an attacker obtains the SSL certificates of Amazon servers and intercepts unencrypted user data by posing to be an Amazon server. Such attacks have been successful in the past enabling attackers to leak client's audio, and video data as well as information related to the user's Wi-Fi credentials. [11] [2]

3.2 End to End Encryption Scheme of Ring Devices

The above shortcomings of Amazon's default encryption scheme can be ameliorated by enrolling in the end-to-end encryption (E2EE) offered by Amazon. E2EE ensures that user data remains encrypted during transmission ensuring that any attacker launching an MITM attack only intercepts encrypted data that cannot be decoded. Amazon has also pushed to make this feature a standard one in their device offerings by slowly phasing out device models that do not support E2EE [7]. In the following paragraphs we discuss the steps involved in E2EE when enrolling a device and when transmitting data to and from the Amazon server.

While E2EE appears to solve all the privacy and security needs of the user, it also comes with a significant reduction in usability. The list of features that users are forced to miss out on simply because they chose to enroll into the E2EE scheme for higher security are listed in Section 3.2.3.

3.2.1 Device Enrollment

Enrolling a Ring device into the E2EE scheme involves two steps - enrolling the primary user mobile device and enrolling the ring device itself. The design consists of secure key-generation, exchange and management between the user's mobile and ring devices.

Enrolling a Mobile Device: The following steps are involved in enrolling a mobile device.

- Passphrase Generation: The Ring app generates a 10-word passphrase locally on the primary mobile device. This passphrase may be required in the future for enrolling additional devices.

- Key Pair Generation: Three asymmetric key pairs are generated locally on the mobile device.

  - Account Signing Key Pair (ASK) - Generated with the RSA-2048 algorithm. Functions as a self signed root of trust to validate the authenticity of other keys.
  - Instance Key Pair (IK) - Generated with the ECC-256 algorithm. The public portion is signed by the ASK and the private portion is in a keystore on the mobile device. It is used for envelope encryption process when encrypting and decrypting videos.
  - Account Data Key Pair (ADK) - Generated with the RSA-2048 algorithm. Used as a backup key for envelope encryption that can be used if the private portion of IK is not available. For example, if a user enrolls a new mobile device into the E2EE scheme and views a video that was encrypted by a prior enrolled device.

- Key Management:

  - The public portion of the IK and ADK are signed by the ASK and uploaded to the Ring cloud.
  - The private portion of the ASK and ADK are encrypted locally and uploaded to the Ring cloud for later use.

The below example illustrates the full flow of enrolling a mobile device into the E2EE scheme.

- A passphrase derived key (PDK) is derived from the 10-word passphrase generated in the passphrase generation stage and a random salt generated with *scrypt* algorithm. This output is used as a password for a Password Based Key Derivation Function (PBKDF2) to derive a key.

- The private portion of the ASK is then locally encrypted with the PDK and uploaded to the Ring cloud.

- The PDK is locally encrypted by the public portion of the IK and uploaded to the Ring cloud. This allows an enrolled mobile device to access the private portion of the ASK to setup the Ring devices without prompting a password.

- The private portion of the ASK is locally encrypted with a new key and stored on the Ring cloud. This new key is generated locally with a 32 byte random value.

- The new key is then locally encrypted with the PDK and then stored on the Ring cloud. This can be later used to unlock the ADK.

- This new key is also locally encrypted with the IK and stored in the Ring cloud. This allows an enrolled mobile device to access the ADK pair.

Now, suppose the user needs to enroll another mobile device into this existing E2EE account and wants to access past videos, the user provides the passphrase in their new device to recreate the PDK. The Ring app then downloads the encrypted private portion of the ASK from the cloud and decrypts with the PDK. The decrypted private portion of the ASK is used to decrypt the new key that was derived from the 32-byte random value. This new key is used to decrypt the private portion of the ADK. This ADK is used to decrypt the videos previously encrypted with the older mobile device. Figure 1 illustrates this entire process with a flow chart.

3.2.2 Viewing a Video Clip

When uploading a video to the Ring cloud or downloading a previously uploaded video for viewing, the following steps are involved.

- The Ring device locally creates an AES-128 symmetric key to encrypt the video produced. This key is generated from scratch each time and is unique to each video.

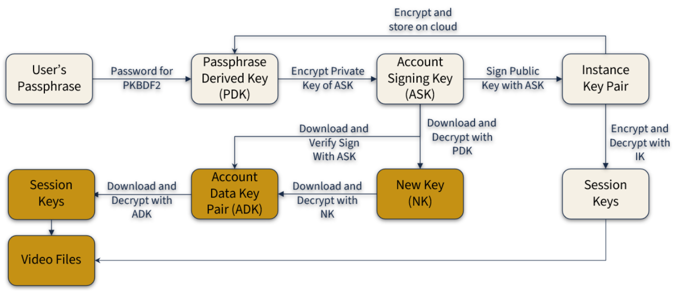
- Then, the device uses envelope encryption to protect the symmetric key. This involves using the public portion of the IK and encrypting the symmetric key with it. This process is also performed with the public portion of the ADK to ensure later enrolled devices can view videos encrypted by older devices.

- These public keys are downloaded from the Ring cloud and validated using the root certificate that was created during enrollment.

- The encrypted symmetric key and related public keys are stored in a manifest file and the video is tagged with this file. The app can later view this file to identify the necessary keys for decryption.

- The encrypted videos and manifest file is uploaded to the Ring cloud.

- At a later time, the Ring app on the registered device can download the video and manifest file. The symmetric key is decrypted with the private portion of the IK found on the device, and then the symmetric key is used to decrypt the final video for viewing.



Encrypt and store on cloud
User's Passphrase
Password for PKBDF2
Passphrase Derived Key (PDK)
Encrypt Private Key of ASK
Account Signing Key (ASK)
Sign Public Key with ASK
Instance Key Pair
Download and Verify Sign With ASK
Download and Decrypt with PDK
Encrypt and Decrypt with IK
Session Keys
Download and Decrypt with ADK
Account Data Key Pair (ADK)
Download and Decrypt with NK
New Key (NK)
Session Keys
Video Files

Figure 1: Flowchart of key management in E2EE scheme

3.2.3 List of Features made unavailable because of E2EE

- Shared users will not be able to view primary user's videos.

- Users will no longer be able to view encrypted videos on the ring.com website.

- Live view feature cannot be used from multiple mobile devices simultaneously.

- Camera previews will not be available on the dashboard.

- Sharing of videos or links is not possible.

- User will not be able to use the Event Timeline feature.

- User cannot view Video Preview Alerts.

- Ring videos cannot be watched on other Amazon or third-party devices

- Quick Replies feature is disabled

- Birds Eye View feature is made unavailable

Feature	**Steps to Enable**
Shared users viewing primary user's videos	Shared users with a data stream key `DSKi` can view all videos uploaded after the key is generated and until it is revoked. Alternatively, they can be granted access for limited period of time through `TDSKi`.
Viewing videos on ring.com website	Configure the browser to be the third-party shared user for your videos thought the `DSK` or `TDSK` process.
View live videos on multiple devices simuletaneously	This feature cannot be replicated as is. However, devices will receive a slightly delayed stream which they can view. The delay is the length of a session and typically only a couple of seconds.
Camera previews on the dashboard	This feature cannot be replicated as is. However, devices will receive a slightly delayed stream which they can view. The delay is the length of a session and typically only a couple of seconds.
Share videos through links	This can be accomplished with some reduced functionality. The primary user needs to download the encrypted video, unencrypt it and re-encrypt it with a new TDSK. This re-encrypted video can be uploaded to Ring and can be shared publicly.
Event Timelines	Share a `DSK` with Ring and add it as a third party user.
Video Preview Alerts	Share a `DSK` with Ring and add it as a third party user.
View videos on other Amazon devices	Share a `DSK` with Ring and add it as a third party user.
View videos on other third-party devices	Share a `DSK` with the device and add it as a third party user.
Quick Replies	This feature can be used without any modifications, the reason for it being disabled in the E2EE scheme is not understood.
Birds Eye View	Share a `DSK` with Ring and add it as a third party user.

Table 1: Features and Steps to Enable Them

4 Proposed Encryption Scheme

Our proposed encryption scheme is based on the idea that encryption should happen at stream granularity instead of account granularity. To this end, we replace the ADK key and its functionalities with separate stream keys. In our new scheme, the ASK signs and authenticates a set of data stream keys (DSKs) which are in turn used to encrypt/decrypt the session keys to view a video or audio stream. This way, Amazon Ring's servers themselves can be treated as a third party with whom data can be shared when convenient. DSKs can also be short lived, in which case they are called temporary data stream keys (TDSKs). TDSKs allow for sharing data only for a window of limited time. The below examples describe the various scenarios where DSKs and TDSKs may be employed. Figure 2 shows the general case of using data stream keys to view video and audio data.

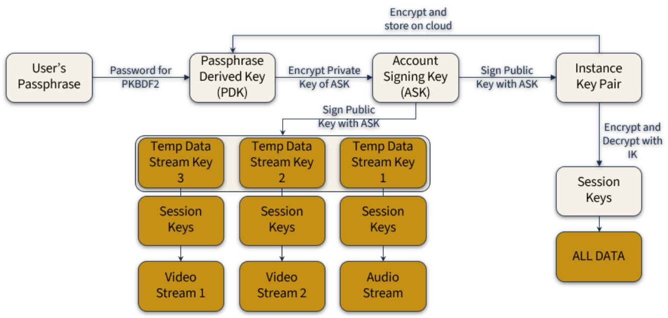


Figure 2: General scheme of using data stream keys to encrypt and decrypt audio and video stream data

4.1 Primary user is not sharing data with anyone

In the simplest case, there exists a primary user who chooses not to share any data with other members. In this case, the primary device of the user encrypts the session keys with the IK. The session keys are used to encrypt and decrypt the data files. All data continues to be stored on Amazon Ring's servers but since Ring does not have access to the DSKs used to encrypt the videos, it cannot view any of the files. This is demonstrated in Figure 3.

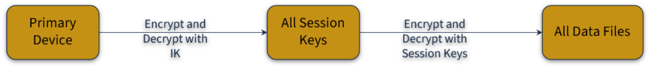

Primary Device
Encrypt and Decrypt with IK
All Session Keys
Encrypt and Decrypt with Session Keys
All Data Files

Figure 3: Simple situation when no data is shared and only primary user has access to all data

4.2 Primary user shares video stream X with user Alice

In this situation, all data is encrypted with session keys which are themselves decrypted with the help of the IK as in the previous case. Additionally, the session key which is used to encrypt the video stream for video X is encrypted with a temporary data stream key TDSK_X. This TDSK_X is shared with user Alice. Alice can now use this data stream key to decrypt the session key for video stream X. Since Alice does not have a way to get access to the other session keys, other data streams are safe and remain private to the primary user. This situation is illustrated in Figure 4.

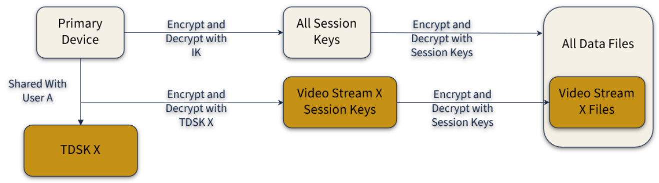

Primary Device
Encrypt and Decrypt with IK
All Session Keys
Encrypt and Decrypt with Session Keys
All Data Files
Shared With User A
Encrypt and Decrypt with TDSK X
Video Stream X Session Keys
Encrypt and Decrypt with Session Keys
Video Stream X Files
TDSK X

Figure 4: Situation when video stream X is shared with Alice

4.3 Primary user shares video stream X with user Alice and Bob and audio stream Y only with user Bob

In this situation, all data is encrypted with session keys which are themselves decrypted with the help of the IK as in the base case. Additionally, the session key which is used to encrypt the video stream for video X is encrypted with a temporary data stream key TDSK_X. This TDSK_X is shared with both users Alice and Bob. Additionally, the session key for audio stream Y is encrypted with stream key TDSK_Y. This TDSK_Y is only shared with user Bob. Alice and Bob can now use this data stream key to decrypt the session key for video stream X. Since Alice does not have a way to get access to the other session keys, other data streams are safe and remain private to the primary user. Bob can use TDSK_Y to decrypt audio stream Y. Since Alice does not have access to TDSK_Y, she has no way to access audio stream Y. This situation is illustrated in Figure 5.

Note here that in our examples we have chosen to illustrate the working of the schemes with temporary data strean keys. This is not a strict requirement. The same mechanism applies to long lived data stream keys. The only difference with DSKs as opposed to TDSKs is that, users who have access to DSKs have access to all streams being added to the account until the DSK is revoked by the primary user.

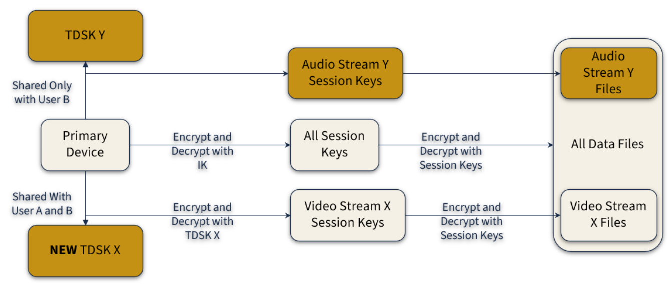

TDSK Y
Shared Only with User B
Audio Stream Y Session Keys
Audio Stream Y Files
Primary Device
Encrypt and Decrypt with IK
All Session Keys
Encrypt and Decrypt with Session Keys
All Data Files
Shared With User A and B
Encrypt and Decrypt with TDSK X
Video Stream X Session Keys
Encrypt and Decrypt with Session Keys
Video Stream X Files
NEW TDSK X

Figure 5: Situation when video stream X is shared with Alice, and Bob. Audio stream Y is shared with Bob

4.4 DSK proliferation

Our scheme necessitates efficient key proliferation. We need to be able to efficiently revoke all pre-existing data stream keys (T)DSK$_i$ when adding a new user to the shared users pool. Once all previous keys are revoked, a new data stream key (T)DSK$_{i+1}$ is shared among all users including the users who were previously part of the trusted user base.

To share DSKs safely with trusted users, we suggest two schemes. Firstly, if physical access to trusted users is available, the primary user can use NFC connections to share the DSKs with other trusted users. In case remote sharing is required, the DSKs can be encrypted with asymmetric keys and shared. This key sharing situation is illustrated in the figure 6.

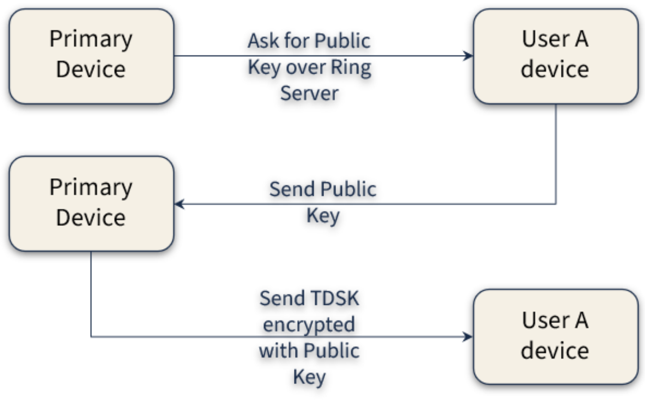

Primary Device
Ask for Public Key over Ring Server
User A device
Primary Device
Send Public Key
Send TDSK encrypted with Public Key
User A device

Figure 6: Setup for primary user to share keys with trusted users

Table 1 details how each feature which was previously unavailable is now made available through our new encryption scheme.

5 Evaluation

To demonstrate the security of the new protocol, we use the Tamarin Prover, a formal verification tool for security protocols [4]. Our system is modelled as a set of rules, and we establish its security properties by proving two key lemmas. Assuming the existing components of the Ring E2EE protocol are secure, we show that the modifications introduced to the system preserve this security. Our proof shows that the creation and sharing of the TDSKs are protected and cannot be obtained by any network adversary.

The proof models secure communication between two users: Alice (the primary user of the Ring account) and Bob (the shared user). Below is a description of all the rules that create the required environment:

- *Register_pk*: Here, a fresh private key (*∼ltkX*) is generated for each user. The private key and its corresponding public key are stored persistently, while the public key is also made publicly available on the network. We assume Ring will provide the public key infrastructure (PKI) to manage and distribute the public keys as needed.
- *Reveal_ltk*: This rule models an adversarial scenario where an agent's private key is revealed on the network.
- *Init_A*: Alice is initialized with her private key (*ltkA*) and Bob's public key (*pkB*). A fresh run ID (*∼id*) is created for this interaction to distinguish parallel runs of the agents. This state is stored as *St_A_1*, representing Alice's starting state.
- *Init_B*: Similarly, Bob is initialized with his private key (*ltkB*) and Alice's public key (*pkA*). His initial state is stored as *St_B_1*.
- *A_send*: Alice generates a fresh nonce (*∼na*) to prevent replay attacks and a new temporary data-stream key (*∼TDSK*). She encrypts these with Bob's public key (*pkB*) and sends the message. The message is also logged with action facts like Send and Secret, ensuring confidentiality and honesty. Alice's state transitions to *St_A_2*, including the nonce and the TDSK.
- *B_receive*: Bob receives the encrypted message and decrypts it using his private key (*ltkB*). After decryption, his state transitions to *St_B_2*, which includes the received TDSK and associated data.

The following lemmas prove key security properties of the protocol:

```
lemma executable:
  exists-trace
    "Ex A B m #i #j. Send(A,m)@i & Recv(B,m) @j"
```

Lemma *executable* ensures the protocol is functional by proving that any message sent by Alice at time i is eventually received by Bob at time j,

```
lemma secret_A:.
  "All n #i. Secret(n) @i & Role('A') @i ==>
    (not (Ex #j. K(n)@j)) |
    (Ex X #j. Reveal(X)@j & Honest(X)@i)"
```

while Lemma *secret_A* demonstrates the secrecy of the TDSK generated by Alice. It ensures these secrets remain unknown to the adversary unless Alice's private key is revealed.

These lemmas have been successfully proven using the Tamarin Prover, ensuring the complete confidentiality of the TDSKs, the authenticity of the messages, and protection against replay attacks, even in the presence of network adversaries.

6 Conclusion

The proposed encryption scheme for IoT devices offers a promising solution to enhance privacy and security while maintaining usability. By introducing data stream keys (DSKs) and temporary data stream keys (TDSKs), the system allows for granular control over data sharing without compromising the end-to-end encryption benefits. This approach addresses many of the limitations present in current E2EE implementations, such as those used by Amazon Ring, enabling users to enjoy a wider range of features while maintaining strong security guarantees. The security of the proposed protocol has been formally verified using the Tamarin Prover, demonstrating its resistance to network adversaries and ensuring the confidentiality of shared keys. However, this verification is limited to the newly introduced components of the system.

To further strengthen the security analysis, future work should focus on extending the Tamarin proof to encompass the entire E2EE protocol, including the existing components of the Ring E2EE system. This comprehensive verification would provide a more robust assurance of the overall system's security properties and help identify any potential vulnerabilities in the interaction between new and existing components. Additionally, implementing this proposed encryption scheme in an open-source IoT project would be a valuable next step. This implementation would not only serve as a proof of concept but also allow for community review and improvement of the protocol.

6.1 Individual Contributions

This project was a collaborative effort involving three contributors who each played crucial roles in its development and execution. All three team members initially focused on analyzing Amazon Ring's documentation to gain a comprehensive understanding of both the default and end-to-end encryption (E2EE) models. This analysis was supplemented by using Wireshark to capture and examine the network packets exchanged between mobile devices and Ring servers, providing valuable insights into the communication protocols. Archit took the lead in proposing the new E2EE communication protocol, while Rohit and Nagendra contributed to its optimization. The final presentation of the project was prepared and delivered by Archit. Rohit concentrated on developing and implementing the Tamarin proof to verify the security properties of the proposed protocol. Nagendra was responsible for compiling and writing the final report, synthesizing the team's findings and contributions into a cohesive document.

References

[1] ASSOCIATES, P. Parks Associates: 27https://www.prnewswire.com/news-releases/parks-associates-27-of-smart-thermostat-owners-report-owning-a-nest-thermostat-301659852.html. [Accessed 12-10-2024].

[2] CIMPANU, C. Ring doorbell vulnerability could have allowed hackers to intercept wi-fi credentials, November 2019. Accessed on December 11, 2024.

[3] ENERTIV.COM. Enertiv — enertiv.com. https://www.enertiv.com/resources/faq/what-is-iot-data-collection#:~:text=IoT%20sensors%20can%20be%20deployed,such%20as%20leaks%20and%20floods. [Accessed 12-10-2024].

[4] MEIER, S., SCHMIDT, B., CREMERS, C., AND BASIN, D. The tamarin prover for the symbolic analysis of security protocols. In *Computer Aided Verification: 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings 25* (2013), Springer, pp. 696–701.

[5] NARCOTTA, J. Strategy Analytics: Amazon's Ring Remained atop the Video Doorbell Market in 2021 — businesswire.com. https://www.businesswire.com/news/home/20220622005023/en/Strategy-Analytics-Amazons-Ring-Remained-atop-the-Video-Doorbell-Market-in-2021, 2021. [Accessed 12-10-2024].

[6] RING LLC. End-to-end encryption. Whitepaper, Ring, July 2021. Accessed on December 11, 2024.

[7] RING LLC. How to set up video end-to-end encryption (e2ee), 2024. Accessed on December 11, 2024.

[8] RING LLC. Privacy, 2024. Accessed on December 11, 2024.

[9] ROUHI, L. Extra layers of security and control, February 2020. Accessed on December 11, 2024.

[10] SINHA, S. State of IoT 2024: Number of connected IoT devices growing 13 https://iot-analytics.com/number-connected-iot-devices/, 2024. [Accessed 12-10-2024].

[11] SPRING, T. Ring doorbell flaw opens door to spying, February 2019. Accessed on December 11, 2024.