AdullamoT: Using IoT Devices as Relays for Time-decoupled Secret Exchange & Censorship Circumvention

Steffen Wendzel
steffen.wendzel@uni-ulm.de
University of Ulm
Ulm, Germany

Abstract

Exchanging policy-breaking or critical information is becoming increasingly challenging due to the expansion of network-level censorship and surveillance by repressive regimes around the world. While typical end-user traffic is often analyzed or blocked by censorship systems, the Internet of Things (IoT) leaves underexplored room for aiding the exchange of confidential information and circumventing censors.

In this paper, we present AdullamoT, a covert channel method that modulates properties of IoT devices to exchange secret messages. Compared to previous work, AdullamoT does not require the modification of IoT devices (i.e., it works without the installation of a covert software on the device) and functions with commercial off-the-shelf (COTS) products. Further, AdullamoT allows a *time-decoupled* exchange, where a sender embeds secret information at one point in time while a receiver fetches the secret information at a later time, which challenges event correlation and forensic analysis. We implement multiple variants using three printers and a smart speaker. In a local WiFi network, our experiments reached mean transmission rates between 14.9 and 349.62 text characters/sec, depending on the device used, and a mean transmission rate of 593.44 text characters/sec when three devices were used simultaneously.

CCS Concepts

• **Security and privacy** → **Malware and its mitigation**; *Pseudonymity, anonymity and untraceability*; *Intrusion detection systems*; • **Applied computing** → *Network forensics*; *System forensics*.

Keywords

covert channels, censorship circumvention, IoT

ACM Reference Format:
Steffen Wendzel. 2026. AdullamoT: Using IoT Devices as Relays for Time-decoupled Secret Exchange & Censorship Circumvention. In *ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '26), June 17–19, 2026, Firenze, Italy*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3785353.3815068

1 Introduction

During the last decade, the number of Internet of Things (IoT) devices grew tremendously. In 2025, an estimated 18.5-21.1 billion devices were connected to the Internet, and it is expected that their number will reach 40.5-52 billion devices by 2034[1,2]. Due to their popularity, numerous IoT devices are accessible over the Internet.

We present AdullamoT (*Adullam*[3] *of Things*), an approach that utilizes IoT devices as covert channel *relays* and *short-term secret data storage*. AdullamoT does not require the installation of software on IoT devices and functions with standard end-user products. Our method allows multiple use-cases, such as censorship circumvention or out-of-band secret data exchange.

Summarized, our key contributions are as follows:

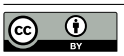
(1) Introduction of AdullamoT, a method for covertly relaying data in a time-decoupled manner. AdullamoT uses IoT devices on the Internet or within local networks to store and exchange secret data.
(2) Provision of a proof-of-concept implementation available to the scientific community.
(3) Evaluation of AdullamoT using four common commercial off-the-shelf (COTS) IoT devices.

The remainder of this paper is structured as follows. We cover related work in Sect. 2 and present our threat scenarios in Sect. 3. Sect. 4 covers our concept, the analysis of selected IoT devices, and our implementation. Sect. 5 conducts an evaluation while Sect. 6 discusses our results. We conclude in Sect. 7.

2 Related Work

Early attempts to transfer and store covert data in IoT/CPS systems were conducted more than 1.5 decades ago. Descriptions and analyzes of illicit information flows in CPS have been provided by Gamage and McMillin [8] (in 2009) as well as Akella *et al.* [1] (in 2010). Wendzel discussed covert channels in CPS (automated buildings) in 2012 [19], and Uluagac discussed sensory channels in CPS [17] in 2014.

Newer works exist as well. Neubert *et al.* [13, 14] show how synthetic secret data can be embedded into industrial control system (ICS) traffic, and Lamshöft *et al.* demonstrate an approach that can be used to exfiltrate covert messages through ICS using long-term process-data storage (sensor measurements). Similarly, Hartmann *et al.* steganographically embed data into configuration databases for ICS, which can be retrieved by a receiver in the future [9]. Moreover, Hildebrandt *et al.* analyze steganography attacks in the context of nuclear plants [10]. *StegFog* by Bieniasz *et al.* is a prototype that allows distributed storage of covert data among multiple nodes which could also be IoT devices running that software [2].



IH&MMSec '26, Firenze, Italy

ACM ISBN 979-8-4007-2376-6/2026/06
https://doi.org/10.1145/3785353.3815068

[1] https://iot-analytics.com/number-connected-iot-devices/
[2] https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/
[3] According to biblical reference 1 Samuel 22:1-2 (also Shmuel I:22 in the Jewish bible), *Adullam* was a hideout place for King David where he also assembled his fellow soldiers. The name AdullamoT was selected since our approach enables a digital hideout by utilizing IoT devices.

Cabaj *et al.* investigate the potential of distributed network covert channels for IoT environments [3] and Cassavia *et al.* analyze their detectability [4].

Our work does not only represent a hiding method. Instead, our main focus is on the security policy-breaking nature of an *indirect* covert channel. Such indirect network covert channels have been studied for decades [24, 25]. Recent work [15] has reviewed a wide range of indirect covert channels, including those that relay covert information through caches (ARP, NTP) of network nodes, which are referred to as *network dead drops*. Given that AdullamoT utilizes IoT device attributes as caches, it can be categorized as such a network dead drop.

A recent work by Kirdan *et al.* [11] realizes indirect covert channels through MQTT. Although several MQTT-based covert channels have been proposed by prior work, their scenario is comparable to ours as their indirect covert channels utilize public MQTT brokers for data exfiltration. An IoT-specific covert data storage (also in the sense of relaying) was proposed by Wendzel *et al.* in 2017 [21]. In contrast to our work, their approach was limited to professional building automation systems running the BACnet protocol. Further, they modulated actuator states and unused registers.

Differences to Previous Work. In contrast to the related work mentioned above, we do not consider ICS, smart cars, energy/smart grid components, professional equipment (e.g., professional printers), or building automation systems. Instead, we utilize widely available low-cost COTS products that are used by typical end-users. Further, AdullamoT does not rely on a specific protocol as most related work (e.g., MQTT or BACnet). In contrast, the communication is done through *some* protocol used by the particular IoT device. Moreover, AdullamoT stores covert data *on* IoT devices *without* deploying any software on these devices. Another major difference from almost all previous work is that AdullamoT creates an *indirect* covert channel that utilizes IoT devices as *relays*. Note that our primary goal is to break a security policy (e.g., the filter policy of a censor), remaining stealthy is a secondary goal. Finally, of our covert channels enable a *time-decoupled* exchange of secret data, i.e., sender and receiver do not need to be active at the same time.

3 Threat Scenarios

AdullamoT can be considered for multiple threat scenarios of which we describe two selected ones.

Internet-based Censorship Circumvention Channel. In this setting, Alice and Bob aim to bypass Internet censorship. Alice is located in a country facing censorship and Bob is located outside of that country. Alice wants to bypass a censoring system that filters traffic leaving the country's national network. To this end, Bob deploys IoT devices accessible through the Internet while Alice exchanges unobtrusive traffic with these IoT devices. Through her exchanges, she influences selected properties of these devices, e.g., the location string of a printer. The censor is expected to apply both, keyword (string) filtering for HTTP traffic, and TLS-based Server Name Indication (SNI) filtering. These techniques (among others) are widely-used filter techniques of today's censors [5, 6, 16, 18, 22]. For this reason, Alice encodes parameters of her traffic in a way that plain keyword filtering does not work (e.g., using Base64 or encryption). Since IoT devices usually contain self-generated certificates and their domain names are unlikely to appear on a censor's list of blocked domain names, TLS-based SNI filtering would not block Alice's flows to the IoT device.

Time-decoupled Dissident's Data Exchange at an Unobtrusive Location. In a setting derived from [21], Alice is a dissident and wants to leak information to Bob who works for an underground news magazine. A repressive government monitors Alice's and/or Bob's behavior and wants to detect potential exchanges of dissident content. Alice and Bob assume that they might be monitored and need to minimize any risk related to their data exchange. For this reason, Alice does not want to meet Bob directly in order not to be seen together and reveal any common exchange of information. As a solution, Alice and Bob appear in a café with a public WiFi. However, they appear at the café at different times, so that Alice leaves a message for Bob that Bob retrieves at another day (and deletes the message after retrieval). They utilize modifiable properties of a local IoT device (printer, smart speaker, ...) located in the café's network to relay secret information.

4 Design & Implementation

4.1 Core Concept

The core idea of our concept is to utilize attributes (or: properties) of IoT devices accessible to both, Alice and Bob, for a covert data exchange. The reason behind this approach is the assumption that the traffic exchange with IoT devices is less focused on by adversaries as these are not part of typical censor blocklists and monitoring.

To this end, Alice and Bob must identify a set D of suitable IoT devices $d_1, \ldots, d_n$ they can somehow influence so that Alice can embed secret messages m of size $s(m)$ into these devices for Bob who extracts these messages. Each message would need to be split into chunks of the maximum provided storage space $s_{max}(d_i)$ of a device.

Minimizing the chance for being blocked would be aided by using multiple IoT devices simultaneously. Thus, if access to *one* device is blocked, access to the *other* devices still remains. We utilize a similar approach like in [20], where we optimized the embedding strategy depending on the network carrier's characteristics. Adapting this concept for our setting, Alice can spread the covert information over multiple IoT devices, depending on their storage space $s_{max}(d_i)$. She makes sure that every IoT device's attributes are used at least sometimes by assigning each IoT device d_i a probability p_i of utilization for the next message chunk, so that $0 < c \leq p \leq 1$ (c serves as a threshold). To this end, she maximizes f (see Eqn. 1).

$$f = \sum_{1}^{n} p_i \cdot s_{max(d_i)}. \tag{1}$$

Optionally, she can define a (repeating) sequence of IoT devices that she shares with Bob, who can retrieve the secret data from the IoT devices.

4.2 Analysis of Covert Storage in IoT Devices

First, we identify suitable attributes of IoT devices that can be utilized to embed covert information. For this reason, we define

two requirements: (1) the attributes should be accessible by typical network devices without a special key or credential, and (2) modifications of the attributes should not be directly recognizable by end-users.

Utilized Devices. We considered $|D| = 4$ low-cost COTS IoT devices listed in Tab. 1, involving three printers and one smart speaker. The table also lists the maximum storage capacity identified for each device $s_{max}(d_i)$ that met our two requirements.

Printers. For both HP printers, we took a look at their configuration websites. These are usually non-protected (it was the case for all three printers although they were taken from environments where they had been used on a regular basis over multiple years). The most suitable option found to embed secret data was to alter the *AirPrint* settings. In the AirPrint settings, one can usually change the name of the printer (string), the device's location (string), and the geographic location (numeric coordinates). Since a change in the printer's name would be easy to recognize by end-users and geographic locations only allow a few bytes of embedding into numeric values, we selected the device location string (s_{max}(HP LaserJet Pro M148dw) = 255 characters and s_{max}(HP LaserJet M15w) = 64 characters) to embed secret data.

Similarly, the Brother printer's web-interface provided a place where contact information could be entered, including a location. We neglected the option to modify other attributes, such as the name of the owner. In contrast to the HP printers, the Brother printer required to handle a CSRF token and HTTPS-connectivity. The provided storage capacity s_{max}(Brother HL – L2375DW) = 100 characters.

Smart Speaker. The Block SB 100 smart speaker runs a Frontier Silicon Internet Radio 2.11-based web interface. The interface did not provide a suitable storage space. However, we identified the combination of HTTP (port 80) and a logging service (port 514) as a suitable carrier: when a user aims to access non-existent URLs through port 80, the logging system on port 514 will report it. Thus, a sender can embed secret data by requesting in-existent files through HTTP while the receiver monitors "not found" reports on port 514. This works only if the receiver filters out irrelevant information provided on the logging port, such as reporting on the *Spotify* streaming service. We experimentally determined that log messages for non-existent files are limited to 59 characters, i.e., s_{max}(Block SB 100) = 59.

Time-decoupled Interaction. If only one message (in case of the printers) or only a few messages (in case of the smart speaker) are sent by Alice, Bob can connect to the IoT device at a later time to retrieve these messages. In case of the printers, the lifetime of secret messages is not limited (Bob could extract the *location* string a year later if he wants to) but the message *length* is limited to $s_{max}(d_i)$ (because Bob would not signal Alice an acknowledgment that allows her to replace the currently embedded message with the next one). In case of the smart speaker, the number of cached log messages is small because a loop buffer is present that shows only the most recent messages ($n = 16$). If new/third-party (e.g., Spotify-related) log messages appear the secret messages would get (partially) overwritten.

4.3 Implementation

We implemented our proof-of-concept tool as a `bash` script for Debian and Ubuntu Linux distributions with limited dependencies, i.e., `openssl`* (used for encrypted HTTP requests), `curl`*, `urlencode`, `nc`, `awk`, and `sed` (*=only used for the Brother printer). The sender scripts transfer HTTP `GET` or `POST` requests to the IoT devices' web services. In case of the printers, the receiver needs to poll for new data and, after receiving a new chunk of data, overwrite previously stored data with a brief acknowledgment message. In case of the smart speaker, the receiver connects to the speaker's port 514 via TCP and parses all data it receives for covert messages.

Code Availability Statement: We made our implementation available: https://github.com/cdpxe/AdullamoT/

4.4 Alternative Communication Relations

In general, our implementation is tailored for an 1:1 communication. For *printers*, an n:1 relation with multiple senders is feasible where these senders could send traffic to the same printer but the acknowledgment would not indicate *which* message was the last one acknowledged, and senders could accidentally overwrite each other's messages before they reach the receiver. A similar problem arises with multiple receivers (1:m) and in multiple senders and receivers (n:m) settings. In such settings, a more complex protocol would be required where every receiver has its own fraction of the acknowledgment message. For example, some characters of a response message could be used to contain a sequence number while the remaining characters could be used for acknowledgments. In such a case, receiver i could use the ith character of the acknowledgment area to indicate its own acknowledgment (response messages are then modified multiple times until all m receivers' acknowledgment characters are set).

In case of the *smart speaker*, an n:1 communication is workable, i.e., multiple senders could send their traffic to a single receiver. This is feasible because the HTTP server accepts GET requests from multiple senders but the logging port can handle only *one* connection at a time. Since there is no acknowledgment mechanism, the receiver collects all messages under the condition that the senders do not exhaust the computing and memory resources of the smart speaker (prevention of message loss).

5 Evaluation

5.1 Evaluation Methodology

We decided to conduct measurements within a local network to reflect the threat scenario of the dissident's data exchange (Sect. 3). Our network connects the IoT devices listed in Tab. 1 using a standard home router. As covert sender and receiver systems we used Debian and Ubuntu-based laptops, running kernel version 6.x. The IoT devices were inactive during all experiments, i.e., they either were turned on but did not face any active utilization or were kept in standby mode. For instance, the smart speaker can be activated through a smartphone app, but its HTTP and system logging services are available for communication independent of its activation status.

Table 1: Utilized IoT Devices for the Experimental Evaluation

Manufacturer, Model	Type	Connectivity	Implemented Protocol	$s_{max}(d_i)$	Mode
Brother HL-L2375DW	laser printer	WiFi	send + wait for simple ACK	100 characters	turned on, inactive
HP LaserJet Pro M148dw	laser printer	Ethernet	send + wait for simple ACK	255 characters	turned on, inactive
HP LaserJet M15w	laser printer	WiFi	send + wait for simple ACK	64 characters	turned on, inactive
Block SB100	smart speaker	Ethernet	blind sending	59 characters	standby mode

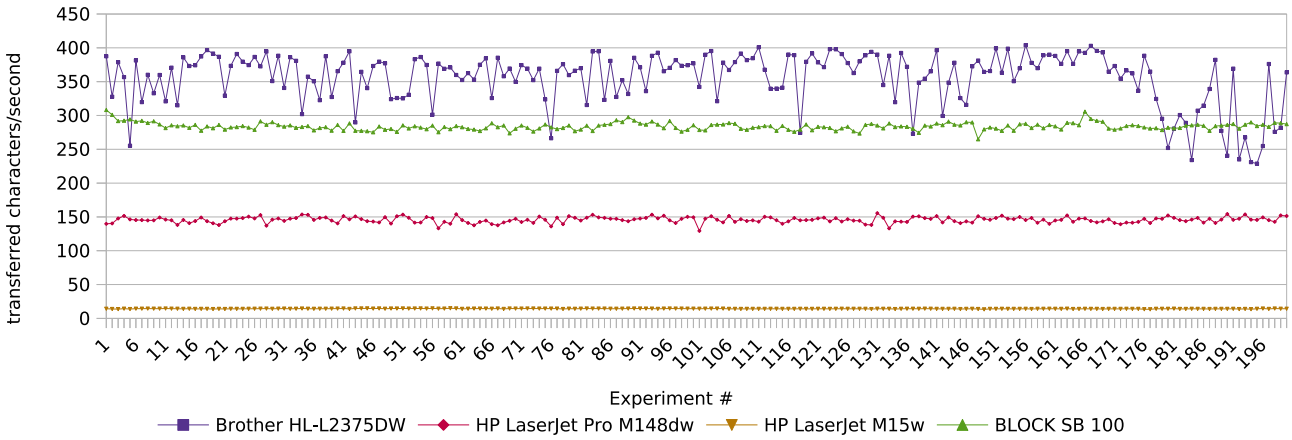


Figure 1: Transmission Rate Per Device During 200 Experiments

5.2 Sending Performance

For our experimental evaluation, we transferred a continuous string of alphabetic characters through our covert channels. We conducted $\geq$ 200 runs for each device, while between 6,400 and 10,000 characters were transferred per run. Fig. 1 shows the sending performance over the first 200 experimental runs. Tab. 2 gives numeric results.

Table 2: Unidirectional Bitrate (Characters/sec)

Manufacturer, Model	Min.	Max.	Mean	σ
Brother HL-L2375DW	202.72	403.98	349.62	45.45
HP LaserJet Pro M148dw	129.32	155.61	145.51	4.26
HP LaserJet M15w	13.34	14.90	14.23	0.24
Block SB100	264.81	308.17	283.85	5.39

As can be seen, the Brother printer allowed for the highest transmission performance (a mean of almost 350 characters/sec, and a peak performance of 403.98 characters/sec), while the HP LaserJet M15w provided the lowest (approx. 14 characters/sec, and a minimum of 13.34 characters/sec). This shows that the performance is highly device-specific. While the Brother printer has provided the highest standard deviation ($\sigma = 45.45$), it has been much lower for all other devices (0.24 to 5.39).

Note that the printer-based covert channels included an acknowledgment feedback channel that the smart speaker covert channel did not foresee.[4] The performance of the feedback channel was *not* included in the plot and table. In other words, the overall performance of our printer-based covert channels is higher than indicated by our measurements as we additionally had to transfer one acknowledgment response for each secret message.

WiFi (instead of Ethernet) connectivity did not appear as a bottleneck: both, the worst (HP LaserJet M15w) and the best performing device (Brother printer) used WiFi (Tab. 1). Also, the standard deviation did not seem to be linked to the WiFi setting as, again, the lowest and highest σ values arose from these two printers.

Multi-channel Transmission Performance. Following the idea of Sect. 4.1 to multiplex our transmission over $\geq$ 2 IoT devices, we performed a parallelized covert message transfer using the three best-performing devices, i.e., the Brother printer, the HP LaserJet Pro printer, and the Block smart speaker. We foresaw 4 bytes/packet for sequence numbers to enable a reassembling of the data at the receiver side (this slightly reduced the available capacity for the actual secret message). We split the transfer so that we distributed the fraction of secret data as follows: 42.1% Brother printer, 35.9% Block smart speaker, and 22.0% HP LaserJet Pro printer, which gave the best performance when all channels were used simultaneously and in a competitive way (all had to go through the same sender and receiver network cards). As shown in Fig. 2, the multi-channel method increased the transmission rate to a mean of 593.44 characters/sec (peak: 664.08 characters/sec).

5.3 Robustness

5.3.1 Immediate Transmission. In our default scenario, Alice relays the data through the IoT device, while Bob extracts the message

[4] Nevertheless, no data loss was experienced for the smart speaker covert channel as the system logging port's TCP stack seemed to buffer the sender-caused messages.

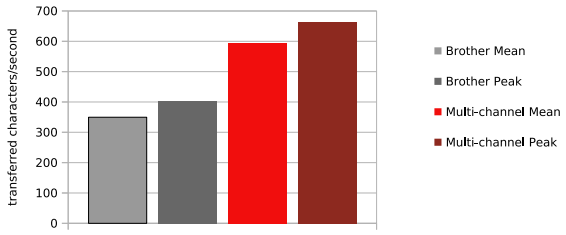


Figure 2: Transmission Rate of the Fastest Printer (Brother) Compared to the Multi-channel Transmission Raate

immediately afterwards (< 10 sec storage duration). The rate of missed and broken covert messages received by Bob was 0% for all printers as well as for the smart speaker. In all printer-related experiments, we employed a trivial protocol between sender and receiver: The sender only stored new data once the last stored data was replaced by an acknowledgment message from the receiver. Both, sender and receiver, constantly monitored the currently stored message through polling, which decreased the throughput. On average, the sender pulled for the receiver's acknowledgment message 2–3 times before sending the next chunk. The receiver pulled for new messages of the receiver between 1-3 times. However, as shown previously (Sect. 5.2), we still achieved a decent performance for all devices with a mean of > 14 characters/sec for the slowest and > 349 characters/sec for the fastest device.

5.3.2 Time-decoupled Transmission (Relaying). In contrast to Bob receiving the messages without delay, one key feature of AdullamoT is its ability to decouple the time of the sending process from the time of the receiving process. This renders it more challenging for an adversary to correlate Alice's and Bob's flows.

For the *printers*, as long as no spurious process[5] or end-user re-configures the device, the delay between sender and receiver can be chosen arbitrarily by Alice and Bob. Note that the used attributes even survive reboots and restarts of the printers.

For the *smart speaker*, we needed to determine the fraction of lost messages over time. The maximum number of cached logging service messages is 16. However, each non-found HTTP resource requested by the AdullamoT sender results in *two* messages that both report the desired *file not found* error. As discussed in Sect. 4.2, we need to cause these errors in order to enable Bob to retrieve the secret messages carried in the reported URLs. However, the fact that two messages are generated means that the maximum number of *cached secret messages* is limited to 8, resulting in a maximum storage space of $8 \times 59 = 472$ characters.

To analyze the time-decoupled relaying through the speaker, we ran an experiment to determine the fraction of successfully received secret data, depending on the storage duration. A key observation is that the provision of secret data through port 514 is not fully reliable. First, we tested a short storage duration of only 1 min, which allowed the receiver to retrieve 100% of the relayed data (tested 130 times). However, when the storage duration increases to 120 min, only 80% of the secret messages were retrieved (tested 25 times). We also tested a setting in which we only embedded secret content into the last 4 (instead of all 8) messages and were able to retrieve 92% of these messages after 120 min.

[5]See Fadlalla [7] for an introduction into the concept of spurious processes.

5.4 Detectability

The main goal of our channel is to break a security policy (e.g., to enable censorship circumvention). However, a secondary goal is to remain undetected. For this reason, we briefly cover the detectability of AdullamoT from different perspectives.

Network-level Detectability. Detecting our covert channel *flows* is feasible as long as anomaly detection or signature-based intrusion detection is deployed, which is not necessarily the case in all settings, such as in a public café. We conducted a simple experiment in which we observed the number of established connections with the HP printer in a real-world setting with three users. Distinguishing from legitimate activity was trivial with a threshold of ≥ 3 established TCP connections per second.

To avoid a trivial detectability of embedded data, the receiver could utilize a printer's *actual* location string as an *acknowledgment* message (this can be configured in our scripts). However, this is not feasible in case of the smart speaker.

Audio-Visual Detectability. Neither the printers nor the smart speaker had to be "used" (e.g., commanded to print papers or to play sound, respectively) during our experiments, i.e., all devices were kept inactive or in standby mode. For this reason, no audible signal was generated by the devices that could be interpreted as an anomaly. In addition, no visual signals, such as blinking LEDs or changes on displays, were observed during our experiments.

Congestion-based Detectability. Modifying location values and causing "not found" errors have a very limited effect on the operation of the IoT devices. However, when Alice and Bob interact with these devices, they consume data capacity and computational power of these IoT devices, which might become noticeable for some cases. In case of the printers, we observed a recognizable delay when print jobs were running while transferring data through our covert channel with the maximum throughput. The delay was in a range of up to 10 sec. For this reason, stealthiness would increase when the covert channel is operated with lower throughput or when printers are unlikely to be used.

6 Discussion

IoT Device Deployment & Ethical Considerations. Alice, Bob, or a person of their choice, could place IoT devices on the public Internet (or any other network), so that they can be used for covert data exchange. While we propose legitimate use-cases, AdullamoT could potentially be misused by attackers utilizing third-party IoT devices. In other words, miscreants could exploit already deployed devices in case they are accessible (e.g., not kept behind NAT or firewalls). We conducted a quick IoT device search using *shodan.io*, which returned 1,160 results for "HP LaserJet Pro", and 4,306 results for "HP LaserJet".[6] Although a large fraction of these devices might not be accessible directly (and an in-depth evaluation was not conducted

[6]The scripts used for the two HP printers are very similar but work flawlessly on both models although their web-interface's appearance differs, and one of the printers is a *Pro* series model. For this reason, we assume that the scripts would work for a larger set of HP laser printers.

due to ethical and legal reasons), we can assume that miscreants would find enough third-party printers to choose from. However, only seven results were found for the Brother model, and none for the BLOCK SB 100. We believe that if methods like AdullamoT are not published while potentially being already known to miscreants, the scientific community could neither develop countermeasures nor provide legitimate users with needed circumvention tools. For this reason, we decided to publish AdullamoT.

Potential Countermeasures. An obvious countermeasure for AdullamoT would be to disable management interfaces of IoT devices or to make them inaccessible through the Internet. Moreover, authentication could be enforced to prevent Alice and Bob from interacting with the devices. However, some devices, such as the smart speaker, do not allow to disable the web interface directly.

Reversibility. Reversibility means that the original message of a carrier can be restored so that it appears as if no secret message was embedded in the past. The *printer*-based channels are *reversible* as the original location values can be restored by the receiver after extracting the secret message. This is done by replacing the secret message with the original one. In contrast, the *smart speaker*-based channel cannot be restored actively by Alice or Bob. Instead, they have to wait for legitimate log messages to overwrite the covert channel's messages after a couple of hours.

Limitations. Our work is linked to the following limitations: (1) We had access to only four devices. Covering a larger number of heterogeneous IoT devices would provide more insights. (2) Our experiments were conducted in a small local area network setting. Internet-based performance metrics might differ. But even if so, the *general* feasibility of our approach is also valid for Internet-based scenarios. (3) We did not consider bypassing NAT'ed environments and we also excluded an investigation of buffering effects when IoT devices are under high workload. (4) Our study lacks an evaluation in which IoT devices are utilized to bypass *real-world* censorship. Such a scenario would require to send traffic from censored countries, eventually considering effects of different routes and regional censorship [12, 22, 23], and is linked to ethical challenges as probe nodes must be deployed in these countries.

7 Conclusion

We introduce AdullamoT, a set of indirect covert channels that utilize commercial off-the-shelf IoT devices as relays for exchanging secret information. Our work aids different use-cases and threat scenarios, especially those benefiting from a *time-decoupled* exchange of secret information. The implementation of AdullamoT is available to the scientific community and has shown a decent transmission rate of up to 403.98 characters/sec using a Brother printer as a relay in a local network.

This paper is considered as a starting point for IoT-based covert channel relaying. Future work can focus on evaluating AdullamoT-like covert channel relays in real-world censorship environments.

References

[1] Ravi Akella, Han Tang, and Bruce M. McMillin. 2010. Analysis of information flow security in cyber–physical systems. *International Journal of Critical Infrastructure Protection* 3, 3 (2010), 157–173. doi:10.1016/j.ijcip.2010.09.001

[2] Jędrzej Bieniasz, Patryk Bąk, and Krzysztof Szczypiorski. 2022. StegFog: Distributed steganography applied to cyber resiliency in multi node environments. *IEEE Access* 10 (2022), 88354–88370.

[3] Krzysztof Cabaj, Piotr Żórawski, Piotr Nowakowski, Maciej Purski, and Wojciech Mazurczyk. 2020. Efficient distributed network covert channels for Internet of things environments. *Journal of Cybersecurity* 6, 1 (2020).

[4] Nunziato Cassavia, Luca Caviglione, Massimo Guarascio, Angelica Liguori, and Marco Zuppelli. 2024. Learning autoencoder ensembles for detecting malware hidden communications in IoT ecosystems. *Journal of Intelligent Information Systems* 62 (2024), 925–949. doi:10.1007/s10844-023-00819-8

[5] Zimo Chai, Amirhossein Ghafari, and Amir Houmansadr. 2019. On the Importance of Encrypted-SNI (ESNI) to Censorship Circumvention. In *9th USENIX Workshop on Free and Open Communications on the Internet (FOCI 19)*.

[6] Anna Crowder, Daniel Olszewski, Patrick Traynor, and Kevin R. B. Butler. 2024. I Can Show You the World (of Censorship): Extracting Insights from Censorship Measurement Data Using Statistical Techniques. In *Annual Computer Security Appl. Conf. (ACSAC'24')*. IEEE, 1123–1138. doi:10.1109/ACSAC63791.2024.00091

[7] Yahia 'Ata Hamid Fadlalla. 1996. *Approaches to Resolving Covert Storage Channels in Multilevel Secure Systems.* Ph. D. Dissertation. University of New Brunswick.

[8] Thoshitha Gamage and Bruce McMillin. 2009. Nondeducibility-based analysis of cyber-physical systems. In *Int. Conf. Critical Infrastr. Prot.* Springer, 169–183.

[9] Laura Hartmann and Steffen Wendzel. 2021. How Feasible are Steganographic and Stealth Attacks on TIA Project Meta-data of ICS: A Case Study with Real-world Data. In *Proc. European Interdisciplinary Cybersecurity Conf. (EICC 2021)*. ACM. doi:10.1145/3487405.3487661

[10] Mario Hildebrandt, Robert Altschaffel, Kevin Lamshöft, Mathias Lange, Martin Szemkus, Tom Neubert, Claus Vielhauer, Yongdian Ding, and Jana Dittmann. 2020. Threat Analysis of Steganographic and Covert Communication in Nuclear I&C Systems. In *Int. Conf. Nuclear Security: Sustaining and Strengthening Efforts.*

[11] Erkin Kirdan and Karl Waedt. 2025. Establishing MQTT Covert Channels Through Public Brokers. In *MILCOM 2025 - 2025 IEEE Military Communications Conference (MILCOM)*. 987–992. doi:10.1109/MILCOM64451.2025.11310520

[12] Xiaoqin Liang, Guannan Liu, Lin Jin, Shuai Hao, and Haining Wang. 2024. Pathfinder: Exploring Path Diversity for Assessing Internet Censorship Inconsistency. *arXiv preprint arXiv:2407.04213* (2024).

[13] Tom Neubert, Bjarne Peuker, Laura Buxhoidt, Eric Schueler, and Claus Vielhauer. 2024. Synthetic Embedding of Hidden Information in Industrial Control System Network Protocols for Evaluation of Steganographic Malware. *arXiv preprint arXiv:2406.19338* (2024).

[14] Tom Neubert, Eric Schueler, Henning Ullrich, Laura Buxhoidt, and Claus Vielhauer. 2025. Extended Analysis, Detection and Attribution of Steganographic Embedding Methods in Network Data of Industrial Controls Systems. *IARIA International Journal On Advances in Security* 18, 1&2 (2025), 2025.

[15] Tobias Schmidbauer and Steffen Wendzel. 2022. SoK: A Survey Of Indirect Network-level Covert Channels. In *Proc. AsiaCCS '22.* ACM, 546–560.

[16] Ram Sundara Raman, Prerana Shenoy, Katharina Kohls, and Roya Ensafi. 2020. Censored Planet: An Internet-wide, Longitudinal Censorship Observatory. In *Proc. 2020 ACM SIGSAC Conf. Computer and Communications Security (CCS '20)*. ACM, 49–66. doi:10.1145/3372297.3417883

[17] A. Selcuk Uluagac, Venkatachalam Subramanian, and Raheem Beyah. 2014. Sensory channel threats to Cyber Physical Systems: A wake-up call. In *Proc. Conf. Communications and Network Security.* 301–309. doi:10.1109/CNS.2014.6997498

[18] Ryan Wails, George Arnold Sullivan, Micah Sherr, and Rob Jansen. 2024. On Precisely Detecting Censorship Circumvention in Real-World Networks. In *Proc. NDSS 2025.* Internet Society. doi:10.14722/ndss.2024.23394

[19] Steffen Wendzel. 2012. Covert and side channels in buildings and the prototype of a building-aware active warden. In *Proc. IEEE ICC'12.* IEEE, 6753–6758.

[20] Steffen Wendzel and Jörg Keller. 2011. Low-attention forwarding for mobile network covert channels. In *Proc. Conf. Commun. and Multimedia Security (CMS 2011) (LNCS, Vol. 7025).* Springer, 122–133. doi:10.1007/978-3-642-24712-5_10

[21] Steffen Wendzel, Wojciech Mazurczyk, and Georg Haas. 2017. Steganography for cyber-physical systems. *Journal of Cyber Security and Mobility* (2017), 105–126.

[22] Steffen Wendzel, Simon Volpert, Sebastian Zillien, Julia Lenz, Philip Rünz, and Luca Caviglione. 2026. A Survey of Internet Censorship and its Measurement: Methodology, Trends, and Challenges. *Computers & Security* 164 (2026), 104732. doi:10.1016/j.cose.2025.104732

[23] Mingshi Wu, Ali Zohaib, Zakir Durumeric, Amir Houmansadr, and Eric Wustrow. 2025. A Wall Behind A Wall: Emerging Regional Censorship in China. In *2025 IEEE Symposium on Security and Privacy (S&P).* IEEE Computer Society, 1307–1324.

[24] Sebastian Zander. 2010. *Performance of Selected Noisy Covert Channels and Their Countermeasures in IP Networks.* Ph. D. Dissertation. Swinburne Univ. of Technology, Melbourne.

[25] Sebastian Zander, Grenville Armitage, and Philip Branch. 2007. A survey of covert channels and countermeasures in computer network protocols. *IEEE Communications Surveys & Tutorials* 9, 3 (2007), 44–57.