Securing Smart Places: A Power-Aware Infrastructure

Tiago Diogo
Instituto Superior Técnico
Av. Rovisco Pais, 1
1049-001 Lisboa, Portugal
tiago.diogo@tecnico.ulisboa.pt

Miguel Pardal
Inesc ID
Rua Alves Redol, 9
1000-029 Lisboa, Portugal
miguel.pardal@tecnico.ulisboa.pt

ABSTRACT

The Internet of Things (IoT) and its vision of connecting every device to one another presents an opportunity to create large information sharing networks. However, intruders can take advantage of the IoT devices constrained nature to disrupt these networks and launch a wide range of attacks on its nodes. In our work we address this issue from a power-aware perspective, trying to find the best relation between security and power consumption. To achieve this objective we do a thoroughly analysis of the existing protocols, attacks and mitigation strategies, combining that information into our proposed smart places network management system. Furthermore, energy consumption profiling was performed to endow future users with the knowledge of what kind of physical resources to deploy, based on the desired network security characteristics.

CCS Concepts

•**Computer systems organization** → **Sensor networks;**
•**Security and privacy** → *Mobile and wireless security;*

Keywords

Internet of Things; Power-Aware Security; Secure Bootstrapping; CoAP; 6LoWPAN; RPL; IEEE 802.15.4

1. INTRODUCTION

The Internet of Things can be seen as a web of interconnected devices that go from everyday wearable objects into fully deployed sensor networks. Despite the huge variety and characteristics of these devices, one thing that they all have in common is the constrained nature that they are built upon. In order to enable the massive deployment to be expected in the near future, IoT devices must be accessible and affordable, capable of operating under lossy wireless networks while being battery powered. This poses a challenge to current Internet protocols since the assumptions regarding the devices' capabilities and objectives do not hold true.

ACSAC '16 Los Angeles, California USA

DOI: 10.475/123_4

To allow the IoT vision to come forward, several new protocols have been developed across the OSI layers, each addressing and tackling the challenges involved in trying to keep the quality and assurances of stronger, more expensive protocols, on constrained systems. After being thoroughly analysed, these protocols have been selected and grouped in a power-efficient stack, establishing a base line for power consumption. Additionally, major attention has been given to information security because for both corporations and individuals, the interconnection of the devices around us can provide information about our choices and whereabouts, therefore leaking corporate information or simply reducing our individual privacy [9]. Thus, the focus moved towards adding mechanism to ensure authentication, confidentiality and integrity of the transmitted information by securing the communication channel. This increased the infrastructure complexity and created the need for a management system capable of storing and flashing network credentials onto new nodes. In order to understand the cost of adding these mechanisms, additional experiments have been performed so that the added power consumption can be measured, profiled and documented, enabling the finding of the best parameters and requirements for a desired level of security.

2. SECURING SMART PLACES

2.1 Protocol Analysis and Selection

.

There are many alternatives and some proposed standards when it comes to choosing a protocol stack for IoT communications. The decision must be based on the particularities of the devices to be used and the objective of the application itself, however a thorough analysis of the existing solutions is a proper way to unveil the strong and weak points of each protocol providing a good basis for an informed decision. A recent survey (January 2015) [1] of IoT enabling technologies, protocols and applications was the starting point for the analysis to follow. The presentation of the available protocols and solutions will follow a bottom-up approach, starting from the data link and physical layer all the way up until the application layer. In particular, the session layer will be left to the end since securing the channel is an optional feature and will be addressed after the application level protocols are properly examined.

2.1.1 Data Link and Physical Layer

.

The first requirement for the physical layer of the IoT is the use of wireless radios. These should aim for simplicity, low-power and low-cost communications. While wireless communication is widespread and can be found from homes to airports, the type of radio commonly used, known as Wi-Fi, uses a high amount of power causing concerns for battery life. In the next paragraphs, an overview of Wi-Fi (IEEE 802.11) is given with the objective of comparing it with the IEEE 802.15.4, a protocol that aims to address these issues.

IEEE 802.11.

.

IEEE 802.11 [3] is a set of standards for Wireless Local Area Networks (WLAN) communications. They are the basis for the so called Wi-Fi. IEEE 802.11 is concerned with high speed, long ranges, message forwarding and high data throughput. These concerns directly clash with the IoT objectives and account for the added power consumption of this protocol.

IEEE 802.15.4.

.

IEEE 802.15.4 [4] on the other hand was created for Low-Rate Wireless Private Area Networks (LR-WPAN) and its specifications focus on low power consumption, low data rate, low cost and high message throughput make it a strong candidate for IoT applications. The IEEE 802.15.4 standard supports two types of network nodes, the Full Function Device (FFD) that acts as coordinator or normal node, and the Reduced Function Device (RFD) that is very simple, with very constrained resources and can only communicate with coordinators. The coordinators are responsible for controlling and maintaining the network. FFD are capable of storing a routing table in their memory and can implement a full Medium Access Control (MAC). IEEE 802.15.4 supports star, peer-to-peer (mesh) and cluster-tree topologies. Regarding performance, it would be unfair to directly compare the two, since IEEE 802.11 transmission power and receiver sensitivity are much greater than 802.15.4. Even if we limit both to a low power level, IEEE 802.11 still outperforms IEEE 802.15.4 in terms of packet delivery ratio, throughput, latency, jitter and average energy consumption. However this comes at the cost of a far lower transmission range [5]. We can conclude that for typical LR-WPAN network requirements, IEEE 802.15.4 is better designed to address the constrained environment issues, while IEEE 802.11 would still be a suitable option if a short transmission range is not a problem.

2.1.2 Network Layer

6LoWPAN.

.

The IoT vision and its massive deployment can only be achieved through the use of IPv6. However, physical layers more suitable for communication over constrained networks pose some limitations to the use of the IPv6 messages. For example, the limited packet size in IEEE 802.15.4 based networks. To tackle these issues, the Internet Engineering Task Force (IETF) IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) [7] working group developed a standard based on header compression to reduce the transmission overhead, fragmentation to meet the IPv6 Maximum Transmission Unit (MTU) requirements and forwarding to link-layer to support multi-hop delivery [2]. 6LoWPAN is able to remove a major share of IPv6 overheads, being able to compress its headers to two bytes, therefore allowing small IPv6 datagrams to be sent over IEEE 802.15.4 networks.

RPL.

.

With the use of 6LoWPAN, upper layer routing protocols can now use the IPv6 addressing scheme. Given the possible frequent topology changes associated with the radio-link instability, successful solutions must take these requirements into account on their specification. Routing Protocol for Low-Power and Lossy Networks (RPL) [10] can support a wide variety of link-layers and is prepared for devices with very limited resources. It is able to build up network routes, distribute routing knowledge among nodes and adapt the topology in a very efficient way. More in depth, RPL creates a Destination Oriented Directed Acyclic Graph (DODAG) between the 6LoWPAN network nodes (Figure 1) that supports unidirectional traffic towards the DODAG root and bidirectional traffic between devices. Each node has a rank that indicates its position relative to other nodes and with respect to the root. This rank is used to create optimized network paths. In order to allow packets to propagate downwards in the topology, either source routing or stateful routing tables are used (More Information on this two types of routing are given in sections 2.2.1 and 2.2.2). For both modes, the DODAG root always maintains a complete list of the network nodes. RPL provides a set of control messages in order to exchange routing graph information. DODAG Information Objects (DIO) are used to advertise information needed to build the DODAG. Destination Advertisement Objects (DAO) are used to advertise information so that downwards traffic can go through the nodes towards the leafs. Nodes may also resort to DODAG Information Solicitation (DIS) messages to request graph information from neighbour nodes. Finally, RPL has a built in topology repair mechanism that acts in the case of a routing topology failure, link failure or node failure. In case the topology needs to be rebuilt, a link layer metric is used to calculate the new route. The new path is considered fit for work if the link layer acknowledgements are received on it.

2.1.3 Application Layer

Hypertext Transfer Protocol (HTTP).

.

HTTP is an application level protocol that uses a request-response model and is the foundation of data communication on the World Wide Web (WWW) It is primarily designed to run over Transmission Control Protocol (TCP) which is a problem in lossy and constrained environments due to the delivery assurances and congestion control algorithms it employs. Besides, HTTP is verbose, text-based, and not suited for compact message exchanges. Moreover, the header size

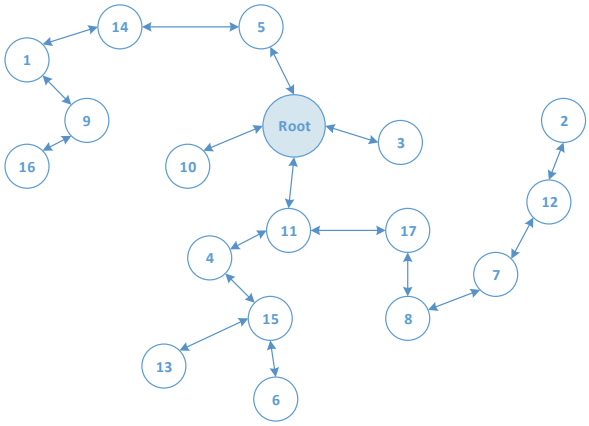


Figure 1: A Sample RPL DODAG.

required for a message exchange can leave too few payload space in constrained networks like the IEEE 802.15.4-based networks where the MTU size of the protocol is 127 bytes. These protocol specifications would not raise any issues in standard WWW communications, but when it comes to constrained environments it is clear that the protocol is not adequate to the necessities of IoT devices and networks.

Constrained Application Protocol (CoAP).

.

CoAP [8] is a document transfer protocol based on REpresentational State Transfer (REST) on top of HTTP functionalities. CoAP objective is to enable tiny constrained devices to use RESTful interactions, where clients and servers expose and consume web services using Universal Resource Identifiers (URIs) together with HTTP Get, Post, Put and Delete methods. Unlike REST, CoAP runs over User Datagram Protocol (UDP) instead of TCP which makes it suitable for full IP networking in small micro-controllers. Retries and reordering are implemented at the application stack using a messaging sub-layer that detects duplicated messages and provides reliable communication using different types of messages. Confirmable messages must be acknowledged by the receiver, nonconfirmable follow the fire-and-forget model. Despite being a lightweight protocol, CoAP still provides important features:

- Resource Observation - CoAP can extend the HTTP request model with the ability to observe a resource therefore monitoring resources of interest using a publish/subscribe mechanism;
- Resource Discovery - CoAP servers provide a list of resources using well-known URIs that allow clients to discover what resources are provided and their types;
- Interoperability - since CoAP is based on the REST architecture, a simple proxy enables CoAP to easily interoperate with HTTP.

.

A study that compared CoAP and HTTP using mobile networks concluded that there is no scenario where CoAP would consume more resources than HTTP [6].

Table 1: Protocol Stack Comparison Overview

Layer	Web	IoT
Application	HTTP	CoAP
Transport	TCP	UDP
Network	IPv6	6LoWPAN
Data-Link/Physical	802.11	802.15.4

2.2 Attack Analysis, Detection and Mitigation

.

Exploitation of existing solutions in the forms of malicious attacks can be found at all the studied OSI layers. They can go from a physical intruder replacing some node on a sensor field to the well-known Denial of Service (DoS) at the application layer. However, given the characteristics of the devices and networks used in IoT combined with the power consumption focus of this work, a specific kind of attacks performed at the network layer is of special interest and importance: battery depletion attacks, also known as, "vampire" attacks.

.

Battery depletion attacks aim at draining the battery, "life", of the network devices, working over time to entirely disable a network, hence being called "vampire" attacks. These attacks do not focus on flooding the network with many packages, instead they drain the node's life by delaying the packets transmission. Many of the existing attacks are not protocol specific [**?**], while others target specific protocols and implementations [**?**]. The following attacks aim at giving an overview of the existing attack possibilities on different routing solutions as well as existing mitigation strategies. Additionally, a range of attacks that target the RPL routing protocol is also analysed. Since RPL is the selected protocol of our energy efficient stack, it is of special importance to consider and assure the mitigation of attacks that would drain the device's batteries by exploiting this light weight protocol's inner workings.

2.2.1 Stateless Protocols

.

In systems that use this type of routing protocols, the source node specifies the entire route to the destination in the packet header. This means that intermediaries do not make decisions regarding the next hop, they only forward to the next node as specified in the original path therefore reducing the amount of computation performed and used energy. However, the source node must ensure that the route is valid at the time of sending and that the neighbour relations among the devices allow the specified forwarding path. Using this transmission scheme, a malicious device can specify paths through the network that are far from optimal, wasting energy at the intermediate nodes who follow the included

malicious source route. The Carousel and Stretch Attacks are examples of these attacks.

Carousel Attack.

.

The objective of this attack is to send a packet along a route composed as a series of loops. This way, a single node may forward the malicious packet several times increasing the total energy consumption by a factor of the number of loops the attacker has introduced on the packet header path. It targets source routing protocols by exploiting the limited verification of the packet headers at the intermediary nodes. Figure 2 shows an example where a vampire node created a path composed of circles around the network when it could have exited after the first hop through the D node.

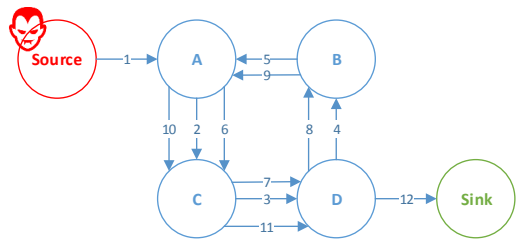


Figure 2: Carousel Attack.

Existing mitigation strategies rely on checking the source route for loops on intermediary nodes, either selecting an appropriate route for the packet or simply dropping it.

Stretch Attack.

.

The objective of this attack is to create a source route around the network, longer than the one that would be required to transverse the network from the source to the sink. The number of elements in the path would be greater than the optimal path, therefore increasing the total energy consumption by a factor of the number of additional hops. Its success rests on intermediary nodes not checking for better paths. Figure 3 shows an example where a vampire node created a path that goes through a greater number of nodes than required to reach the sink.

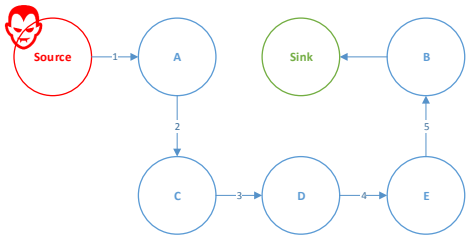


Figure 3: Stretch Attack.

A limited way of mitigating this attack would be to ensure that path routes have less than the total number of devices on the network. Vasserman and Hoper proposed a property called “no-backtracking” that assures the packet is always moving closer to the sink on every hop [**?**].

2.2.2 *Stateful Protocols*

.

In systems that use this type of routing protocols, network nodes are aware of the network topology and its state, being able to make local decisions on the node to whom they will forward the packet. The effect of the Vampires on this type of routing is limited since the route is built dynamically from many independent forwarding decisions. However, attackers can still cause damage by forcing packet forwarding through nodes that would not be on the optimal path, for example, by forwarding the packet back to the source. The Directional Antenna and Wormhole Attacks are examples of these attacks.

Directional Antenna Attack.

.

In this attack, the attacker takes the role of an intermediary and not the source of a packet. If the attacker has the resources to use a directional antenna, it can deposit a packet on arbitrary parts of the network while also forwarding the packet locally. This causes nodes that were not on the optimal path to also consume energy by forwarding a packet they would not normally receive, therefore increasing the total energy consumption by a factor of the directions the attacker can position the antenna and the distance between the receiver and the sink. Figure 4 shows an example where a "vampire" intermediary deposited a node on a distant location of the network, causing the packet to follow two different routes towards its destination

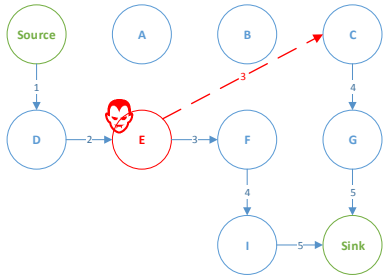


Figure 4: Directional Antenna Attack.

A mitigation strategy could be to analyse the route paths of a given packet that reached the sink more than once. The last node identifier to appear duplicated before the path started to diverge would be one who then directed the packet to multiple regions, therefore revealing the attacker.

Wormhole Attack.

.

This attack can be seen as variation of the Directional Antenna Attack but with the collaboration of two or more attackers. Instead of simply forwarding the packets to arbitrary parts of the network, the attacker emulates a link between them and advertises to the network that recently formed connection. This disrupts the topology and has severe impact on routing paths since attackers can indicate that the link cost between them is very low, and therefore influence the forwarding decisions of neighbour nodes. By using these malicious routes, the energy consumption is increased because either this channel does not exist at all (packets are dropped and need to be resent), or the transmission cost between the attackers is greater than the normal message propagation through the network. Figure 5 shows an example where two vampires emulate a connection between them influencing the routing decisions of their neighbours. The hops numbered with prime numbers represent the path taken by a packet after the wormhole is constructed. Although the packet still reached the destination, the cost of the wormhole path is greater than the previous regular path.

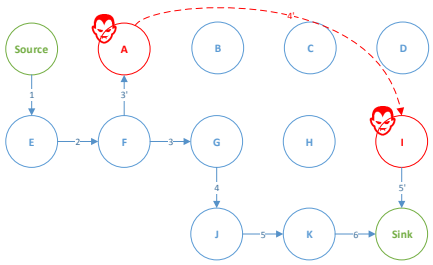


Figure 5: Wormhole Attack.

Wormhole attacks can be prevented using the Merkle tree authentication [**?**]. This tree is organized from the leafs towards the root where every parent knows their children and asks them for authentication based on their ID and public key.

2.2.3 RPL Specific Attacks

.

Selective Forwarding Attack.

.

In a selective forwarding attack, a malicious node can launch a DoS attack by selectively forwarding packets. Its main goal is to disrupt routing paths but can be used to filter any protocol. Since RPL has built in topology repair mechanisms, a full packet filtering would trigger a healing phase and leave the malicious node out of the topology. For sustainability, an attacker could let the RPL control messages pass by and drop the remaining packets. Depending on the routing scheme being used (source routing or stateful tables) the source could first verify path availability or each node could dynamically decide to forward the packet through another path with similar quality. In any case, a good approach would be to report those failures to the underlying RPL system in order to trigger a preventive healing and improve the route quality.

Hello Flooding Attack.

.

The Hello in the name of this attack comes from the initial message a node sends when joining a network. By broadcasting this message with a strong signal power, an attacker can try to introduce himself as neighbour to many nodes of the network, or at least force a large portion of the network to spend energy starting the message exchange for node insertion. A simple solution for this attack would be to test the bi-directionality of the link. If no acknowledgement is received, the path is discarded. Another approach, if geographical locations of the nodes are known, would be to discard every hello message coming from a location beyond the transmission capabilities of ordinary nodes.

2.2.4 Protocol Independent Attacks

.

The last addressed category is not dependant on network topologies or protocol messages. It focuses on attacks that can be performed regardless of the used protocol and whose goal is to obtain information about a network device. With that information an attacker can, for example, try to include himself in the network as a legitimate device or spoof his identity to forward traffic towards him. The Clone and Sybil Attacks are examples of these attacks.

Clone ID and Sybil Attack.

.

As the name suggests, in a clone ID attack, the attacker steals the identity of a legitimate network node by copying the information of that node onto another node. This way the attacker can gain access to the traffic that was destined to the legitimate node, prevent packets to reach their intended destination and can even influence voting schemes. The Sybil attack is similar to the Clone ID, with the difference that the attacker uses several stolen identities on the same physical node. This way, large parts of a network can be taken over without the need to deploy several physical nodes. Figure 6 shows an example of a clone ID attack where the cloned attacker received the packet that was originally destined to the legitimate node.

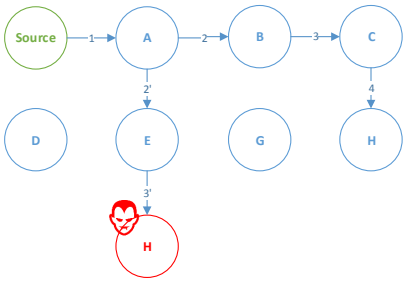


Figure 6: Clone ID Attack.

Proposed mitigation strategies for this type of attacks consist on keeping track of the number of instances of each identity. By using the node neighbours, either a centralized or distributed approach could be used to detect duplicate entries.

2.3 Secure Bootstrapping

.

The term bootstrapping is applied to the process in which a new device is connected to an existing network. To achieve a secure bootstrapping, a unique identity and security parameters are associated with the device during this phase. There are several ways to carry out the initial setup, either via a physical interface or wirelessly. In the case of wireless bootstrapping, attention must be given to eavesdropping so that the secure credentials cannot be intercepted.
Since many of the studied attacks are to be performed by a malicious intruder capable of interacting with the network, if we could assure a secure bootstrapping, meaning that the new node would be authenticated before becoming an active member of the network, a large portion of those attacks could no longer be performed. The following bootstrapping techniques were summarized in [**?**] and aim at providing secure bootstrapping for IoT devices.

2.3.1 Token-Based

.

In token-based distribution, device specific security credentials are generated and written to a token. That token can range from memory sticks or flash cards to Radio Frequency Identification (RFID) tags or smartcards. Is has the advantage that this initial credential generation can be performed on a physically controlled environment and only later, on the commissioning phase, is the token plugged into the device. After the successful insertion of the security credentials, the token can be removed and collected back into the secure environment. This process can be considered of high security since the credentials are generated on a closed

environment and are transmitted through a physical link. To further increase the security level, a password could be used to encrypt the credentials, however, that would require the device to have some kind of interface that would allow to input the password. In the case of a large number of devices, this approach would be unsuitable due to the management effort of manually deploying the tokens to the devices [**?**].

2.3.2 *Identifier-Based Access Control List*

.

With an identifier based Access Control List (ACL), new devices are allowed or denied access to the network based on their unique ID. A commonly used identifier is the MAC address. This has some major drawbacks in security since, firstly, it provides no assurances on the first time the device connects to the network. An attacker can easily intercept the first messages and get access to the device information. And secondly, after the bootstrapping phase, MAC addresses can be spoofed by an attacker, allowing him access to the network by bypassing the ACL with the identifier of a legitimate node.

2.3.3 *One-Time-Passwords*

.

The use of one-time-passwords enhances the manual input of credentials on the device to be bootstrapped. The person responsible for the deployment of the new node should receive through a secure channel an one-time-password, that would then be used to authenticate the node, by authenticating its locally generated key material. This material can be either a certificate request to a Certificate Authority (CA) or a locally generated public/private key pair. The achieved security level is proportional to the security of the channel used to obtain the one time password, but assuming that channel is secure, so is this method. The drawback is that it forces devices to possess some kind of interface to insert the one time password.

2.3.4 *Manufacturer Installed Credentials*

.

So far, excluding the identifier based access control list, the intent of the studied techniques is to supply to the new device the security credentials needed to obtain access to the network, or at least provide an authentication method that allows fetching those credentials. In manufacturer installed credentials, those security credentials are deployed during the manufacturing process of the device vendor. Those credentials are typically a public/private key pair certificate bound to the identifier of the device. This certificate can be integrated into the initial loading of the firmware or stored in a separate integrated circuit designed for credential storing. In the second case, this method's security can be considered very high since those integrated circuits assure that the private key cannot be read from memory. This way, the new device comes shipped with the necessary security credentials not only for the bootstrapping phase but also for the normal operation phase since it does not need to fetch any additional credentials. The effort is on the root or management station that needs to import the vendor CA certificates to assure the new device credentials are trustworthy. Also the production costs increase, implying an increased device cost.

2.4 System Architecture

A numbered display equation – one set off by vertical space from the text and centered horizontally – is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n \to \infty} x = 0 \tag{1}$$

Notice how it is formatted somewhat differently in the **displaymath** environment. Now, we'll enter an unnumbered equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

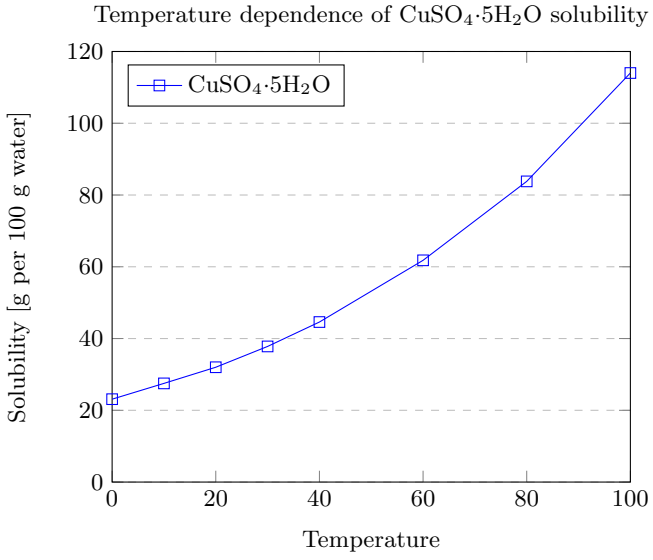
$$\sum_{i=0}^{\infty} x_i = \int_{0}^{\pi+2} f \tag{2}$$

just to demonstrate LaTeX's able handling of numbering.

3. EVALUATION

In this section we measure and profile the resources required for the system operation. This ranges from the memory required for the nodes operative system and protocol stack, to power consumptions and battery replacement expectations. Each following subsection both presents the collected data and explains the process and technologies involved in obtaining it.

3.1 Hardware Requirements



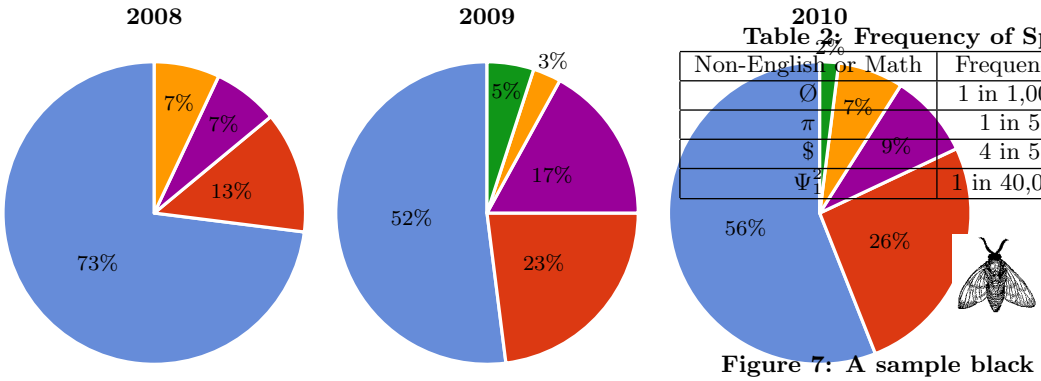
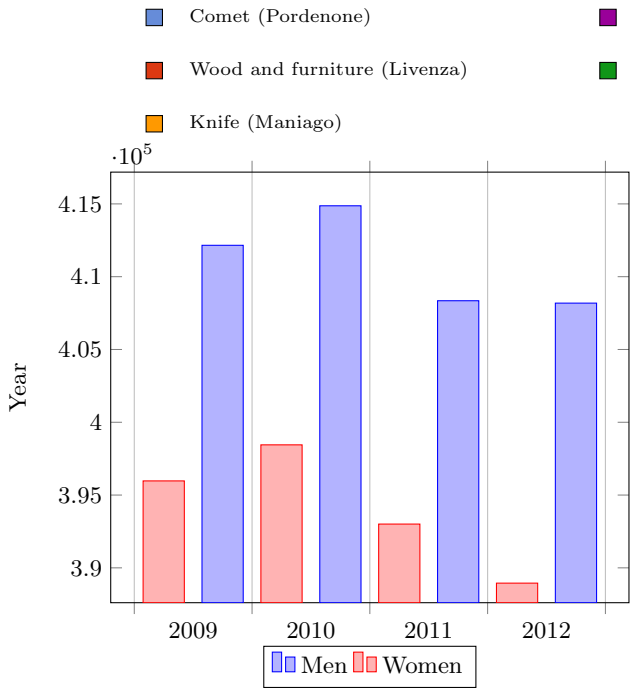


Table 2: Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

Figure 7: A sample black and white graphic.



Citations to articles [?, ?, ?, ?], conference proceedings [?] or books [?, ?] listed in the Bibliography section of your article will occur throughout the text of your article. You should use BibTeX to automatically produce this bibliography; you simply need to insert one of several citation commands with a key of the item cited in the proper location in the `.tex` file [?]. The key is a short reference you invent to uniquely identify each work; in this sample document, the key is the first author's surname and a word from the title. This identifying key is included with each item in the `.bib` file for your article.

The details of the construction of the `.bib` file are beyond the scope of this sample document, but more information can be found in the *Author's Guide*, and exhaustive details in the *LaTeX User's Guide*[?].

This article shows only the plainest form of the citation command, using `\cite`. This is what is stipulated in the SIGS style specifications. No other citation format is endorsed or supported.

3.2 Hardware Requirements

Because tables cannot be split across pages, the best placement for them is typically the top of the page nearest their initial cite. To ensure this proper "floating" placement of tables, use the environment **table** to enclose the table's contents and the table caption. The contents of the table itself must go in the **tabular** environment, to be aligned properly in rows and columns, with the desired horizontal and vertical rules. Again, detailed instructions on **tabular** material is found in the *LaTeX User's Guide*.

Immediately following this sentence is the point at which Table 1 is included in the input file; compare the placement of the table here with the table in the printed dvi output of this document.

To set a wider table, which takes up the whole width of the page's live area, use the environment **table*** to enclose the table's contents and the table caption. As with a single-column table, this wide table will "float" to a location deemed more desirable. Immediately following this sentence is the point at which Table 2 is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed dvi output of this document.

3.3 Power Consumption

Like tables, figures cannot be split across pages; the best placement for them is typically the top or the bottom of the page nearest their initial cite. To ensure this proper "floating" placement of figures, use the environment **figure** to enclose the figure and its caption.

This sample document contains examples of **.eps** files to be displayable with LaTeX. If you work with pdfLaTeX, use files in the **.pdf** format. Note that most modern TeX system will convert **.eps** to **.pdf** for you on the fly. More details on each of these is found in the *Author's Guide*.

As was the case with tables, you may want a figure that spans two columns. To do this, and still to ensure proper "floating" placement of tables, use the environment **figure*** to enclose the figure and its caption. and don't forget to end the environment with figure*, not figure!

Figure 8: A sample black and white graphic that has been resized with the `includegraphics` command.

Table 3: Some Typical Commands

Command	A Number	Comments
`\alignauthor`	100	Author alignment
`\numberofauthors`	200	Author enumeration
`\table`	300	For tables
`\table*`	400	For wider tables

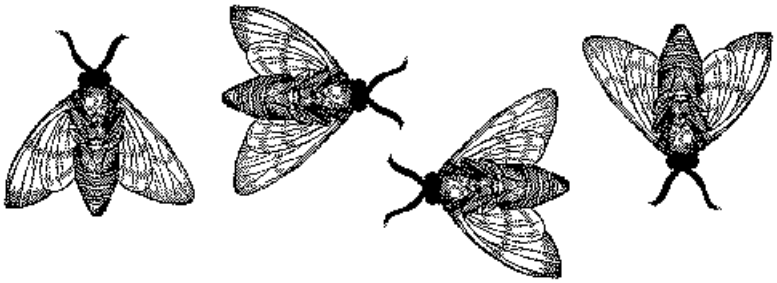


Figure 9: A sample black and white graphic that needs to span two columns of text.

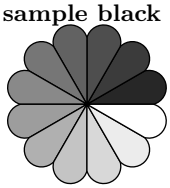


Figure 10: A sample black and white graphic that has been resized with the `includegraphics` command.

3.4 Theorem-like Constructs

Other common constructs that may occur in your article are the forms for logical constructs like theorems, axioms, corollaries and proofs. There are two forms, one produced by the command `\newtheorem` and the other by the command `\newdef`; perhaps the clearest and easiest way to distinguish them is to compare the two in the output of this sample document:

This uses the **theorem** environment, created by the `\newtheorem` command:

THEOREM 1. *Let f be continuous on $[a, b]$. If G is an antiderivative for f on $[a, b]$, then*

$$\int_a^b f(t)dt = G(b) - G(a).$$

The other uses the **definition** environment, created by the `\newdef` command:

Definition 1. If z is irrational, then by e^z we mean the unique number which has logarithm z:

$$\log e^z = z$$

Two lists of constructs that use one of these forms is given in the *Author's Guidelines*.

There is one other similar construct environment, which is already set up for you; i.e. you must *not* use a `\newdef` command to create it: the **proof** environment. Here is a example of its use:

PROOF. Suppose on the contrary there exists a real number L such that

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = L.$$

Then

$$l = \lim_{x\to c} f(x) = \lim_{x\to c} \left[gx \cdot \frac{f(x)}{g(x)}\right] = \lim_{x\to c} g(x) \cdot \lim_{x\to c} \frac{f(x)}{g(x)} = 0 \cdot L = 0,$$

which contradicts our assumption that $l \neq 0$. □

Complete rules about using these environments and using the two different creation commands are in the *Author's Guide*; please consult it for more detailed instructions. If you need to use another construct, not listed therein, which you want to have the same formatting as the Theorem or the Definition[**?**] shown above, use the `\newtheorem` or the `\newdef` command, respectively, to create it.

A *Caveat* for the TEX Expert

Because you have just been given permission to use the `\newdef` command to create a new form, you might think you can use TEX's `\def` to create a new command: *Please refrain from doing this!* Remember that your LATEX source code is primarily intended to create camera-ready copy, but may be converted to other forms – e.g. HTML. If you inadvertently omit some or all of the `\def`s recompilation will be, to say the least, problematic.

4. CONCLUSIONS

This paragraph will end the body of this sample document. Remember that you might still have Acknowledgments or Appendices; brief samples of these follow. There is still the Bibliography to deal with; and we will make a disclaimer about that here: with the exception of the reference to the LATEX book, the citations in this paper are to articles which have nothing to do with the present subject and are used as examples only.

5. ACKNOWLEDGMENTS

This section is optional; it is a location for you to acknowledge grants, funding, editing assistance and what have you. In the present case, for example, the authors would like to thank Gerald Murray of ACM for his help in codifying this *Author's Guide* and the **.cls** and **.tex** files that it describes.

6. REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. *IEEE Communications Surveys & Tutorials*, PP(99):1–1, 2015.

[2] J. Hui and D. Culler. Extending IP to low-power, wireless personal area networks. *IEEE Internet Computing*, 12(4):37–45, 2008.

[3] IEEE. *IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, volume 2012. 2012.

[4] IEEE Computer Society. *Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, volume 2011. 2011.

[5] N. K. N. Kok Seng Ting, Gee Keng Ee, Chee Kyun Ng and B. M. Ali. The Performance Evaluation of IEEE 802 . 11 against. (October):850–855, 2011.

[6] T. Savolainen, N. Javed, and B. Silverajan. Measuring Energy Consumption for RESTful Interactions in 3GPP IoT Nodes. pages 1–8, 2014.

[7] Z. Shelby, S. Chakrabarti, E. Nordmark, C. Systems, C. Bormann, and Ericsson. Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). https://tools.ietf.org/html/rfc6775, 2012.

[8] Z. Shelby, K. Hartke, and C. Bormann. The Constrained Application Protocol (CoAP). https://tools.ietf.org/html/rfc7252, 2014.

[9] A. Ukil, S. Bandyopadhyay, and A. Pal. Privacy for IoT: Involuntary privacy enablement for smart energy systems. *2015 IEEE International Conference on Communications (ICC)*, pages 536–541, 2015.

[10] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. https://tools.ietf.org/html/rfc6775, 2012.

APPENDIX

A. HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. In the **appendix** environment, the command **section** is used to indicate the start of each Appendix, with alphabetic order designation (i.e. the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure *within* an Appendix, start with **subsection** as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

A.1 Introduction

A.2 The Body of the Paper

A.2.1 Type Changes and Special Characters

A.2.2 Math Equations

Inline (In-text) Equations.

Display Equations.

A.2.3 Citations

A.2.4 Tables

A.2.5 Figures

A.2.6 Theorem-like Constructs

A Caveat for the TEX Expert

A.3 Conclusions

A.4 Acknowledgments

A.5 Additional Authors

This section is inserted by LaTeX; you do not insert it. You just add the names and information in the `\addition-alauthors` command at the start of the document.

A.6 References

Generated by bibtex from your .bib file. Run latex, then bibtex, then latex twice (to resolve references) to create the .bbl file. Insert that .bbl file into the .tex source file and comment out the command `\thebibliography`.

B. MORE HELP FOR THE HARDY

The sig-alternate.cls file itself is chock-full of succinct and helpful comments. If you consider yourself a moderately experienced to expert user of LaTeX, you may find reading it useful but please remember not to change it.