Ad Hoc Networks 000 (2018) 1–11

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Efficient DCT-based secret key generation for the Internet of Things

George Margelis[a], Xenofon Fafoutis[a,b,*], George Oikonomou[a], Robert Piechocki[a], Theo Tryfonas[a], Paul Thomas[a]

[a] Faculty of Engineering, University of Bristol, UK
[b] DTU Compute, Technical University of Denmark, Denmark

ARTICLE INFO

Article history:
Received 8 September 2017
Revised 22 June 2018
Accepted 18 August 2018
Available online xxx

Keywords:
Secret key generation
Physical layer security
IoT Security
Internet of Things (IoT)
IEEE 802.15.4

ABSTRACT

Cryptography is one of the most widely employed means to ensure confidentiality in the Internet of Things (IoT). Establishing cryptographically secure links between IoT devices requires the prior consensus to a secret encryption key. Yet, IoT devices are resource-constrained and cannot employ traditional key distribution schemes. As a result, there is a growing interest in generating secret random keys locally, using the shared randomness of the communicating channel. This article presents a secret key generation scheme, named SKYGlow, which is targeted at resource-constrained IoT platforms and tested on devices that employ IEEE 802.15.4 radios. We first examine the practical upper bounds of the number of secret bits that can be extracted from a message exchange. We contrast these upper bounds with the current state-of-the-art, and elaborate on the workings of the proposed scheme. SKYGlow applies the Discrete Cosine Transform (DCT) on channel observations of exchanged messages to reduce mismatches and increase correlation between the generated secret bits. We validate the performance of SKYGlow in both indoor and outdoor scenarios, at 2.4 GHz and 868 MHz respectively. The results suggest that SKYGlow can create secret 128-bit keys of 0.9978 bits entropy with just 65 packet exchanges, outperforming the state-of-the-art in terms of energy efficiency.

1. Introduction

As the Internet of Things (IoT) becomes part of our every day lives, more physical objects are interconnected and remotely controllable through the Internet. This paradigm introduces new security risks, allowing malicious users to gain access to objects and information that are traditionally considered secure [1]. In addition, a large number of these objects are connected using wireless technology, which makes the communications vulnerable to eavesdropping. This highlights the need for confidentiality, which is typically realised through encryption schemes.

The challenge is that such wireless embedded devices are typically severely constrained in terms of computational power, memory, energy, and hardware space. Hence, traditional security services, found in the upper layers of the OSI model, *e.g.* encryption protocols based on certificate management and key distribution [2], are not applicable for resource-constrained IoT devices. As a result, there is research interest in creating resource-efficient security protocols that can supplement and support lightweight cryptographic schemes [3–5].

An interesting alternative to key distribution is generating secret bit sequences on the devices using channel observations. These bit sequences can be used directly as an encryption key or as a seed to a number generator. This approach is based on the theory of reciprocity for electromagnetic propagation, which suggests that the channel between two transceivers is symmetrical within the coherence time. Indeed, several works propose methods that enable two communicating parties to generate an encryption key, leveraging the channel impulse response during bidirectional transmissions [6,7]. The Mutual Information (MI) of the channel observations by the two transceivers defines the maximum key generation rate.

Prior theoretical works studied the MI of in a variety of channels and calculated its theoretical upper bound [8–10]. Other experimental works employ the Received Signal Strength (RSS) and measure its entropy to estimate the size of the extracted bit sequence shared between the transceivers. In practice, however, the extracted bit sequences have errors, and these works rely on error correction schemes, such as Forward Error Correction (FEC), to mitigate any inconsistencies. As a step forward, measuring the MI

[*] Corresponding author.
E-mail address: xefa@dtu.dk (X. Fafoutis).

https://doi.org/10.1016/j.adhoc.2018.08.014
1570-8705/© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)

of the RSS observations provides practical upper bounds and quantifies the need for error correction.

In this article we first measure the MI of channel observations in several practical scenarios using IEEE 802.15.4 transceivers at 2.4 GHz and 868 MHz. Because of the nature of the transceivers, we use the RSS of each received packet as a channel observation. After calculating the practical upper bounds of the secret key generation rate, we present SKYGlow: a secret key generation protocol that is designed for low-power IoT platforms. Different to the existing work, SKYGlow operates in the frequency domain. Indeed, the proposed scheme employs a Discrete Cosine Transform (DCT) stage, and we use experimental data, collected using the aforementioned transceivers, to verify its enhanced performance. In addition, we quantify the probability that an eavesdropper can reconstruct the secret key. In summary, the contributions of this work are:

- In a series of practical scenarios, we measure the MI of the common channel observations between two communicating transceivers, assuming the existence of an eavesdropper. We consider two cases: with knowledge and without knowledge of the channel observations by the eavesdropper. The former is used to quantify the maximum key generation rate, whilst the latter is used to quantify the maximum key generation rate in secrecy from the eavesdropper.
- We present a new secret key generation scheme, named SKYGlow, whose performance is verified on experimental data. We compare the performance of SKYGlow to similar schemes in the literature, also designed for IoT applications, as well as against the previously calculated practical upper bounds.
- Extending our previous work [11,12], this study considers two different scenarios that cover a wide range of IoT deployments: an indoor deployment operating at 2.4 GHz, and an outdoor deployment operating at 868 MHz. In these scenarios we consider both the cases of static and mobiles nodes.

The remainder of the paper is structured as follows. Section 2 covers the prior work, including efforts to assess the limits of MI for each observation, as well as other secret key generation schemes targeting the IoT domain. Section 3 presents the results of our investigation on the practical upper bounds of the secret key capacity. Section 4 elaborates on the proposed scheme, SKYGlow. Section 5 evaluates SKYGlow, presents insights regarding its advantages and disadvantages, and compares it against other secret key generating schemes designed for the IoT. Finally, Section 6 concludes the article.

2. Prior work

Prior work has investigated the idea of generating shared random keys at two transceivers, using the shared randomness of the common communicating medium. Some pioneering work in this field is done by Ahlswede and Csiszár [9,10], who looked into the generation of shared randomness by two communicating terminals in secrecy from a third party. These works define the term 'key capacity' as the maximum secret key generation rate by two terminals that observe correlated sources, and prove that it is equal to the MI, assuming a Discrete Memoryless Multiple Source Model (DMMS).

These early works were extended in [8], which employs the MI to quantify theoretical upper bounds for the shared randomness of ultra-wideband channel observations. Other related works examine the generation of keys from observing various radio parameters [7,13–15]. Yet, these works are not directly applicable to IoT devices because they either have a purely information-theoretic perspective, or make use of specialized hardware for extracting the observations. Different to these works, this article studies the practical MI between two communicating parties, using solely off-the-shelf IoT devices.

Since the publication of [9], several works present key generation algorithms with various results [13,16–19]. Assuming that Alice and Bob wish to generate a secret shared key, these algorithms generally incorporate the following steps:

1. Firstly, Alice and Bob communicate unencrypted messages to each other.
2. Alice and Bob collect observations that capture the effect of the channel on each received message. In the case of SKYGlow, this corresponds to the measurement of the exchanged messages' RSS, as shown in Fig. 4.
3. The collected RSS values form a time-series that is quantized and transformed into a binary series.
4. In practice, each message exchange is not simultaneous as half-duplex transceivers are used. The time delay between receiving a message and transmitting a response is responsible for discrepancies between Alice's and Bob's binary series. These errors are reconciled, either using error correction [5,20] or some reconciliation protocol, such as *Cascade* [17]. In [21], the authors opt for sacrificing entropy to eliminate errors.
5. Finally, after the two sequences become identical, they usually have low entropy or, in the case of the Cascade protocol, significant information has leaked to the eavesdropper in the process. Thus it is necessary to transform the sequences, in a way that increases the entropy of the key and obfuscates any partial information may have leaked to an eavesdropper during key reconciliation. This process, commonly referred to as *privacy amplification*, leads to sequences with reduced size.

SKYGlow works similarly, but operates in the frequency domain. Indeed, SKYGlow introduces a DCT stage before quantization. The goal of SKYGlow is to produce a secret bit with every RSS value, in contrast to previous works where a significant portion of the observations is unused. This yields to a reduction of the needed radio usage (receptions and transmissions), and increases the energy efficiency of key generation. When reliability is more important than energy efficiency, SKYGlow can be tuned to a higher key generation probability, by trading off the amount of secret bits generated per packet exchanged. Finally, the proposed scheme results in keys with very high entropy, without the need of the privacy amplification stage. Thus, the energy consumption associated with this processing stage is avoided, and the size of the sequence is not reduced. Both contribute to the efficiency of SKYGlow.

3. Practical limits of the key generation rate

The MI, $I(X;Y)$, between X and Y is expressed as:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X), \tag{1}$$

where

$$H(X) = -\sum_{x \in X} p(x)\log_2 p(x), \tag{2}$$

$$H(Y) = -\sum_{y \in Y} p(y)\log_2 p(y), \tag{3}$$

and

$$H(X|Y) = -\sum_{y \in Y}\sum_{x \in X} p(x,y)\log(p(y|x)). \tag{4}$$

The probabilities $p(x)$ and $p(y)$ are calculated empirically from the measured frequencies of x and y.

The MI corresponds the key capacity, defined as the maximum achievable key generation rate [9]. In this article the random variables are RSS measurements for each exchanged frame. Thus, X and

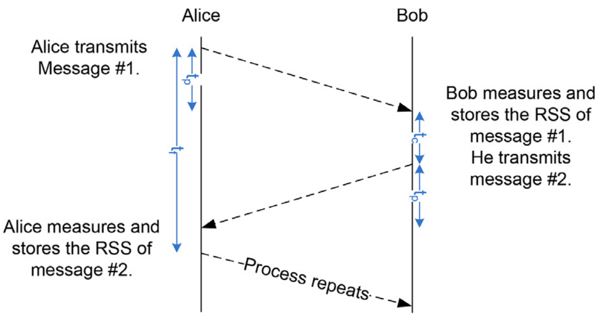

Alice
Bob
Alice transmits Message #1.
Bob measures and stores the RSS of message #1. He transmits message #2.
Alice measures and stores the RSS of message #2.
Process repeats

Fig. 1. The methodology of RSS logging.

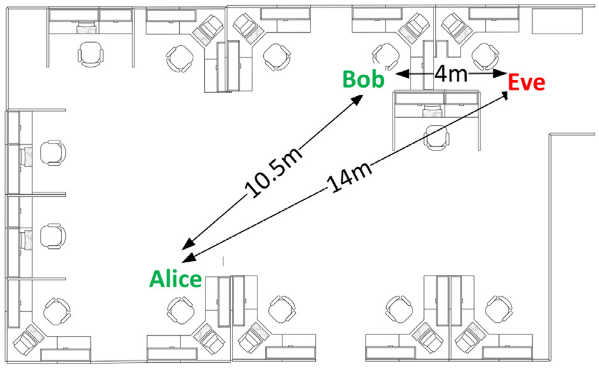


Fig. 2. The topology of the indoor scenario.

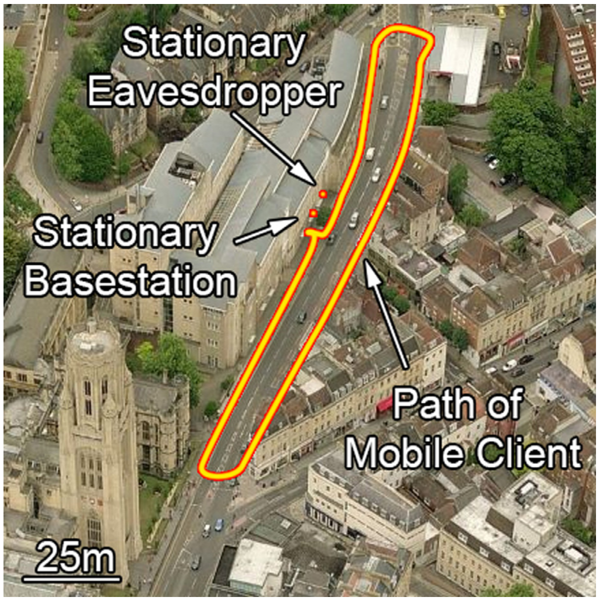


Fig. 3. The topology of the outdoor scenario.

Y correspond to the RSS measurements of Alice (A) and Bob (B), respectively. It is noted that, due to hardware constraints, the granularity of the measurements is a single RSS value per frame.

Because of the reciprocity of the channel, we expect X and Y to be correlated, and the MI to be greater than zero, $I(X;\, Y) > 0$. Furthermore, (1) highlights the dependency of the MI on the entropy of X and Y, indicating that high-entropy environments can eventually translate to higher key generation rates.

3.1. Secret key generation rate

The existence of an eavesdropper, Eve (E), may have a negative effect on the key generation rate. Let us consider that Eve eavesdrops all the exchanged messages. We denote as $z_x \in Z_x$ and $z_y \in Z_y$ the recorded RSS values for X and Y respectively. According to Wilson et al., [8], the secret key generation rate is bounded by:

$$K(X;Y||Z) = \min[I(X;Y), I(X;Y|Z_x), \\ I(X;Y|Z_y), I(X;Y|Z_xZ_y)]. \tag{5}$$

We highlight that, in practice, the eavesdropper may miss some of the exchanged messages due to packet loss. However, in this article, we assume that Eve can receive all of the packets, as we want to focus on the worst case scenario.

3.2. Threat model

In this section, we define Eve's abilities. The first assumption is that the eavesdropper can listen to all communications between Alice and Bob, *i.e.* the legitimate communicating parties. We also assume that Eve is separated from Alice and Bob by more than the coherence distance [22]. In addition, we assume that Eve does not use her radio to inject traffic or jam the channel. We also assume that Eve eavesdrops the channel with a single-radio/single-antenna system. In future work, we intend to loosen this assumption and consider the case of an omnipresent Eve [23]. We do not make any further assumption on her hardware capabilities. Before the communication link between Alice and Bob is cryptographically secured, we assume that Eve is able to capture the unencrypted syndrome, sent to Bob for reconciliation, and use it to reconstruct the secret key. After the first key is established however, we assume that all subsequent syndromes are encrypted before transmission.

3.3. Implementation, measurements and experimental results

We implemented our system in Contiki, the open source operating system for IoT devices. For the indoor measurements we use the CC2650 radio [24] that operates at 2.4 GHz. For the outdoor measurements we use the CC1310 radio [25] at 868 MHz. In both cases, three devices are used, acting as Alice, Bob, and Eve. These devices have half-duplex radios; hence, the captured RSS sequences correspond to messages transmitted with a small time delay. The message exchange sequence is shown in Fig. 1, where the transmission delay is denoted as t_p; t_c is the time delay between the reception of the packet and transmission of the response; and t_f is the time between two successive iterations of the process. We highlight that t_c must be as small as possible, to ensure the reciprocity of the channel. In our implementation, $t_p = 2.4$ ms, $t_c = 7.8$ ms and $t_f = 1$ s for the indoor case; and $t_p = 12$ ms, $t_c = 7.8$ ms and $t_f = 0.2$ s for the outdoor case.

We performed a variety of experiments to measure the MI in scenarios with different attributes. Fig. 2 shows the locations of Alice, Bob, and Eve in an indoor scenario (office space). In this setting, we consider various scenarios with or without Line of Sight (LoS) between the devices in stationary positions. In addition, we consider the case of Alice being mobile, whilst Bob and Eve are in their stationary positions. It is noted that the channel is very dynamic, as Alice is moving inside and outside of the room. Thus, we expect mobility to create high variance in the RSS sequences.

Fig. 3 depicts the outdoor location of Alice, Bob, and Eve. Similarly, we positioned Alice and Bob in opposing sides of a busy road next to the University of Bristol. The constant traffic assures that the channel is as dynamic as is typical to an urban environment. All of the devices are on street level, and occasionally LoS is established. Eve is positioned 1.5 m away from Bob, with LoS, stationary and without any objects to disrupt the LoS. Thus, the RSS from Bob's messages for Eve remains constant. In addition, we also consider the case of Alice being mobile, following the highlighted path.

Please cite this article as: G. Margelis et al., Efficient DCT-based secret key generation for the Internet of Things, Ad Hoc Networks (2018), https://doi.org/10.1016/j.adhoc.2018.08.014

Table 1
Summary of the measurements. X corresponds to Alice, Y corresponds to Bob, and Z_x and Z_y are the sequences of RSS observations by Eve respectively. The secret key generation rate is marked in bold.

Scenario	$\min(H(X), H(Y))$	$I(X; Y)$	$I(X; Y \mid Z_x)$	$I(X; Y \mid Z_y)$	$I(X; Y \mid Z_x, Z_y)$
Indoors: Alice, Bob and Eve with LoS	3.638	2.500	2.490	2.503	**2.473**
Indoors: Alice, Bob with LoS, Eve without LoS	3.634	2.566	2.546	2.470	**2.424**
Indoors: Alice, Bob and Eve without LoS	3.868	2.850	2.678	2.821	**2.592**
Indoors: mobile scenario #1	4.541	3.128	3.079	3.063	**2.918**
Indoors: mobile scenario #2	4.833	3.307	3.357	3.396	**3.051**
Outdoors: stationary with occasional LoS	3.937	2.136	1.490	2.097	**1.512**
Outdoors: mobile scenario #1	5.554	2.787	2.818	2.833	**2.804**
Outdoors: mobile scenario #2	5.620	2.596	2.667	2.671	**2.642**

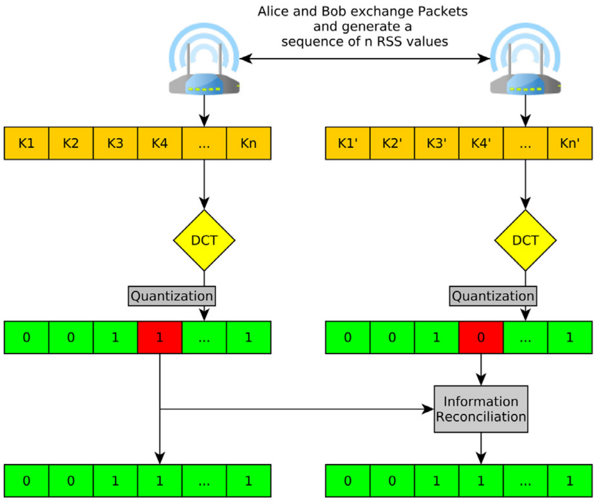


Fig. 4. Overview of SKYGlow.

The results are summarized in Table 1. The second column provides the minimum entropy of the sequences of RSS values logged by Alice and Bob. Assuming that the two sequences had perfect correlation, this would also be the mutual information of the two sequences. Since correlation is not perfect, the mutual information seen in the third column of the table, is actually a fraction of the maximum possible; for indoor scenarios, I(X; Y) varies from 68 to 75% of the possible maximum, while for outdoor scenarios it varies from 46 to 54% of the possible maximum, a result explained by the even more reduced correlation of the measurements in the outdoor environment. Regardless, because of the Data Processing Inequality, we cannot increase I(X; Y) further than the values of the third column, which are our practical upper boundaries and can be used to assess the performance of any method to derive secret bits from similar application scenarios.

Furthermore, we note that the maximum key capacity is always higher than 2.1 bits, while the secret key capacity is generally higher than 2.4 bits, and never less than 1.5 bits. Secret key generation schemes for IoT devices in the existing literature generate at best 1 secret bit per exchange, as we can see in Section 5. Hence, there is still considerable space for improvement.

4. SKYGlow

This section presents SKYGlow, providing further details on each of its stages. As highlighted before, SKYGlow generates keys of high entropy without requiring a privacy amplification stage. In this article, we present two versions of the proposed algorithm; SKYGlow.a for applications that prioritize energy efficiency, and SKYGlow.c for applications that prioritize reliability. The algorithms share the same principles, with the only variations being found in the DCT phase, which is our most important contribution. An overview of SKYGlow is shown in Fig. 4.

4.1. Sampling and DCT

The first stage of SKYGlow is RSS sampling. The protocol targets single-radio low-power IoT devices. Such devices typically provide the RSS indicator, which indirectly captures the channel attenuation. Most commercial radios do not provide detailed information on how the given RSS indicator is calculated. Yet, we assume that the process is identical in both devices.

SKYGlow applies a DCT to the series of RSS values, before feeding them to the quantizer. The DCT converts a finite time-series of samples (*i.e.* the RSS values) into a sum of cosine functions at different frequencies:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \qquad k = 0, \dots, N \tag{6}$$

As shown in the next section, SKYGlow outperforms other related key generation schemes in energy-efficiency. This advantage originates from the introduction of the DCT stage, which curates the input signal, enabling higher efficiency at the quantization stage. Efficient adaptability and resilience to bit injections are additional advantages, as briefly discussed in the remainder of this section. Fig. 5 illustrates the DCT.

The DCT stage makes the proposed scheme able to dynamically discard high frequency components that are largely responsible for bit errors. Thus, SKYGlow is tunable, and able to increase the rate when the channel is more symmetrical, and decrease the rate when there is a lower degree of reciprocity (for example, when the coherence time is lower than the sampling period).

We take advantage of this feature to tailor the performance of SKYGlow to the application scenario, introducing two versions of the algorithm, namely SKYGlow.a and SKYGlow.c. SKYGlow.a applies the DCT detailed in (6) with N equal to the number of RSS values collected and then passes those N cosine wave components to the quantizer detailed in the next section. SKYGlow.c discards the higher $N/2$ cosine wave components, and passes to the quantizer only the lower frequency components. These lower frequency components have increased correlation compared to the ones with higher frequencies. We note however that, as we will see later, this increase leads to a reduction in the amount of bits generated per message exchange. An illustration of the difference between SKYGlow.a and SKYGlow.c can be seen in Fig. 6.

We note that although the results of SKYGlow.c are similar to passing the RSS vector through a low pass filter before passing the values to the quantizer, our scheme generates all cosine amplitudes in advance, and can then fine-tune the number of components needed, without having to go through the process again. In contrast, with a low pass filter the process would have to be started again from the beginning, costing time and energy. Furthermore, our scheme allows an overlaying adaptation layer to dynamically discard different number of DCT components under specific conditions. For example, Alice and Bob may choose to drop a certain amount of high frequency components after n failed key generation attempts. Moreover, as we will see later, were SKYGlow

Please cite this article as: G. Margelis et al., Efficient DCT-based secret key generation for the Internet of Things, Ad Hoc Networks (2018), https://doi.org/10.1016/j.adhoc.2018.08.014

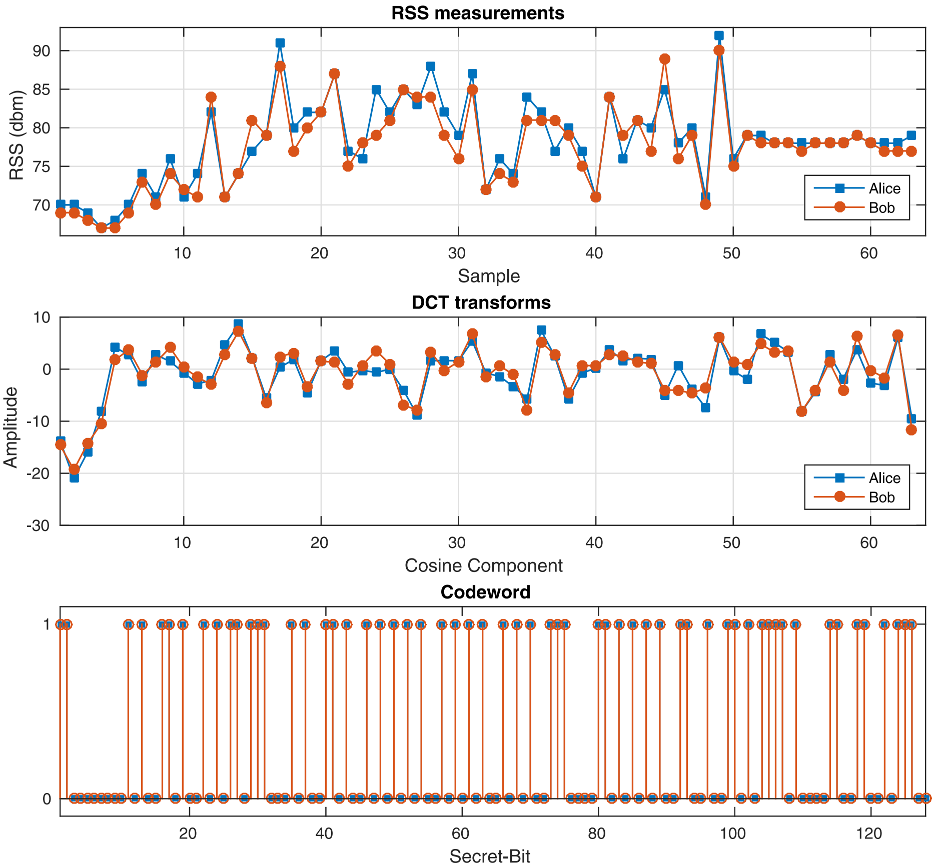


Fig. 5. An example of a series of 64 RSS measurements (top). The output of the Discrete Cosine Transformation (middle). The generated 128-bit secret key (bottom).

to be implemented in a wearable device, it could employ SKYGlow.a in an indoor environment but SKYGlow.c when moving outdoors, where the channel is much more dynamic and correlation degrades. The transition could take place seamlessly without any need for additional hardware.

There is also a security advantage that stems from the DCT stage. One important vulnerability of schemes similar to the one described in [13] is that they are vulnerable to attacks that can lead to bits injected by a malicious actor following the same method as the one suggested in [26]. To summarize this method, Eve measures the RSS of both the eavesdropped messages transmitted from Alice and Bob, and measures the difference between them. When the difference is minimal or even non-existent, that means that the channel between Eve and Alice and the channel between Eve and Bob are very highly correlated, and thus a multicast message transmitted inside the coherent time of the channels would be received from both Alice and Bob with the same RSS, because of the reciprocity of the channels. By adjusting the transmitting power of those messages and spoofing the identity of the sender, Eve could potentially inject bits in the bitstream that Alice and Bob generate, that are highly correlated and thus persist through the information reconciliation stage. For a more elaborate description of the method, we refer the reader to [26].

Schemes like the ones described in [13,16] are vulnerable to such an attack. However, the DCT stage makes this attack less practical. Consider, that due to slow and fast fading, it is not possible in most practical environments for Eve to predict when the channels between Eve & Alice and Eve & Bob will be highly correlated, thus Eve cannot predict when she can inject bits. More importantly though, the injected bits are not directly fed to the quantization stage to be converted to 0s or 1s. Instead the DCT stage creates a set of coefficients from a set of bits. Even if a number of them are injected, because Eve has no knowledge of the remaining she would not be able to predict how those injected packets affected the overall time series, and thus it would be more challenging to predict the effect on the coefficients. Alice and Bob, on the other hand, proceed as usual.

4.2. *Quantization*

In the next stage, SKYGlow quantizes x_n and uses the quantized output to generate a secret bit sequence. Most related works use a quantizer with a censoring region (*e.g.* [21]). Such quantizers operate as follows. A censoring region, $[\mu + \alpha\sigma, \mu - \alpha\sigma]$, is defined by a parameter $\alpha > 0$, where μ is the mean and σ is the standard deviation of the input series, and all values that are in this region are discarded. The samples $x_n > \mu + \alpha\sigma$ are then quantized as 1 and the samples $x_n < \mu - \alpha\sigma$ are quantized as 0. This approach filters the samples that have a low degree of correlation, due to *e.g.* thermal noise, out of the key generation process. Employing a censoring region has two drawbacks. Firstly, if a message exchange is very close to the border of the censoring region, there is a probability that only one of the two communicating parties will include it. This results to desynchronized sequences and, thus, further errors. Secondly, the discarded measurements are a source of wasted energy. Indeed, the larger the censoring region, the more packets are exchanged without contributing to the generated key. In an energy-conscious IoT context, such inefficiencies are undesirable.

The quantizer of SKYGlow calculates the mean, μ, and the standard deviation, σ, of the cosine components, provided by the DCT stage. The quantizer does not have a censoring region, instead it

Please cite this article as: G. Margelis et al., Efficient DCT-based secret key generation for the Internet of Things, Ad Hoc Networks (2018), https://doi.org/10.1016/j.adhoc.2018.08.014

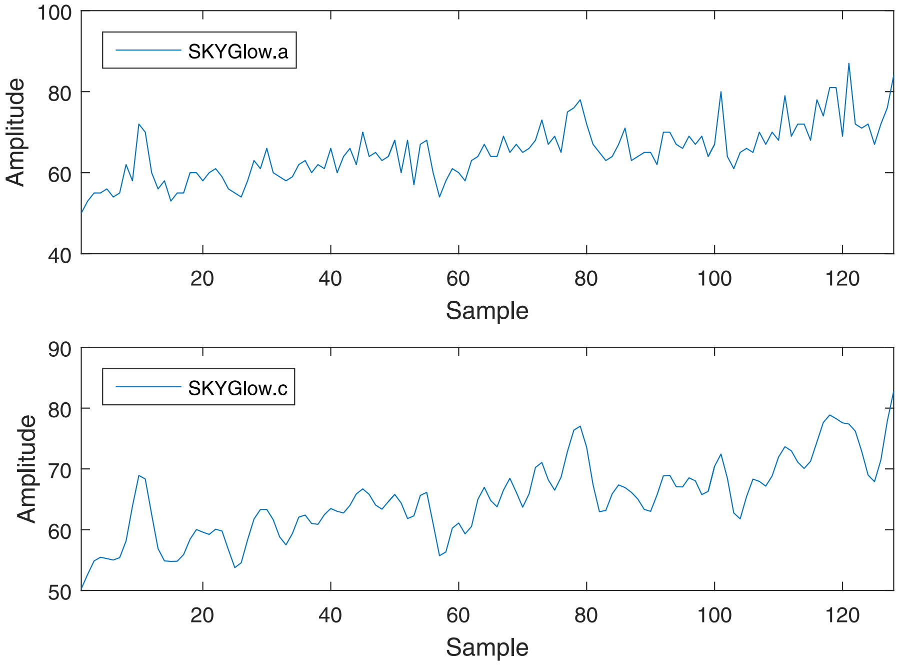


Fig. 6. The inverse transform of the output of the DCT stage of SKYGlow.

operates as follows, generating two bits per cosine component:

$$Q(n) = \begin{cases} 11, & x_n \geq \mu + \sigma, \\ 10, & \mu \leq x_n < \mu + \sigma, \\ 01, & \mu - \sigma < x_n < \mu, \\ 00, & x_n \leq \mu - \sigma. \end{cases}$$

4.3. Error correction

The channel is not perfectly symmetrical, due to the time delay between messages and the effect of thermal noise. These leads to potential inconsistencies between the RSS series of Alice and Bob. These inconsistencies can, in turn, lead to differences in the cosine waves, and, if close to the quantization thresholds, to bit mismatches.

In [21], the authors sacrifice entropy to reduce the mismatches with filtering. Other works employ Cascade [27]. Yet, Cascade is unsuitable for energy-constrained IoT devices, at it requires a large number of message exchanges to operate [28].

As mentioned in Section 4.1, SKYGlow is tunable and able to efficiently discard the high frequency cosine waves, in accordance with the characteristics of the environment. Any bit mismatches at the quantization stage are addressed with the Information Reconciliation stage, as shown in Fig. 4. This stage is implemented using Slepian-Wolf Low Density Parity Codes (LDPC) with a code rate of 7. This configuration ensures that the syndrome can be transmitted with a single 802.15.4 packet. Hence, SKYGlow only needs to send one additional packet to support error correction.

5. Evaluation

SKYGlow.a and SKYGlow.c are evaluated using the measurements described in Section 3.3. SKYGlow.a generates a single 128-bit key with every set of 64 observations (*i.e.* 65 packets including the syndrome). Similarly, SKYGlow.c operates on sets of 128 observations. As a result, SKYGlow.a can generate up to 1.96 secret bits per packet and SKYGlow.a can generate up to 0.99 secret bits per packet. In practice however, due to mismatches that our coding cannot correct, the long term average secret bits per packet are expected to be lower than the maximum values.

5.1. Evaluation metrics

The performance of SKYGlow is assessed with the following list of metrics:

- Bit Error Rate (BER): BER denotes the bit mismatch probability between two generated keys.
- Key Agreement Rate (KAR): KAR denotes the probability of generating identical keys with no bit errors.
- Key Leakage Rate (KLR): KLR denotes the probability of Eve reconstructing the key using the unencrypted syndrome.
- Key Entropy: The generated key should be random, thus, with an entropy that is ideally 1.
- Secret Bits per Packet (SBP): SBP denotes the number of generated bits per message exchange between the communicating parties. SBP is a proxy for energy-efficiency.
- Energy per 128-bit Key (EK-128): EK-128 denotes the energy required to generate a 128-bit key. Smaller values indicate a more energy-efficient system that is more suitable for energy-constrained IoT devices.

If the BER is high, LDPC decoding may fail to correct the mismatches. Let us denote the bit success probability as p, *i.e.* KAR is equal to $1-p$, and the number of packets needed for the generation of one key as N. On average, $E[N]$ packets are required for generating a single key, as given by:

$$E[N] = \sum_{n=1}^{\infty} N p^{n-1}(1-p)n = \frac{N}{1-p} \ ,$$

where $N = 65$ or $N = 129$ for SKYGlow.a and SKYGlow.c respectively. $E[N]$ is also employed as a performance metric that, similarly to SBP, acts as a proxy for energy-efficiency.

To maximize security, it would be natural to execute SKYGlow periodically and generate fresh keys. As a result of the fact that the very first syndrome is unencrypted, the first key would be more vulnerable than the following ones. Assuming that q is equal to the KLR, we calculate the expected number of leaked packets as:

$$E[M] = \sum_{m=1}^{\infty} q^m \cdot E[N] = \frac{q}{1-q}\frac{N}{1-p}$$

Please cite this article as: G. Margelis et al., Efficient DCT-based secret key generation for the Internet of Things, Ad Hoc Networks (2018), https://doi.org/10.1016/j.adhoc.2018.08.014

Table 2
Performance metrics for indoor scenarios at 2.4 GHz.

	SKYGlow.a			SKYGlow.c		
	Stationary		Mobile	Stationary		Mobile
	LoS	nLoS		LoS	nLoS	
BER (A)	0.032	0.007	0.035	0.001	0.001	0.003
BER (E)	0.483	0.483	0.485	0.476	0.482	0.480
KAR	0.835	0.897	0.844	0.902	0.911	0.938
KLR	0.007	0.010	0.004	0.009	0.008	0.008
SBP	1.64	1.77	1.90	0.89	0.90	0.93
Entropy	0.997	0.998	0.997	0.998	0.998	0.998
E[N]	77.66	72.34	67.36	142.90	141.50	137.46
E[M]	0.52	0.73	0.28	1.29	1.41	1.10
EK-128	36.3	33.8	39.5	66.8	66.1	64.3

$E[M]$ is employed as a metric that evaluates security.

EK-128 can also be approximated in a similar manner. Assuming that Q_{tx} and Q_{rx} is the electric charge required to transmit and receive a packet respectively, and that Q_p is the electric charge required for executing SKYGlow, EK-128 is given by

$$\text{EK-128} = \sum_{n=1}^{\infty} p^{n-1}(1-p)V\Big(N(Q_{tx}+Q_{rx})+Q_p\Big)n \ ,$$

where V is the system voltage. Due to the significant improvements in energy-efficiency of modern microcontrollers [29], the fact that DCT can be hardware-accelerated [30], and the fact that communication occurs two orders of magnitude more frequently than processing (*i.e.* by factor of N), for the remainder of the section we consider that radio dominates the energy required to generate a key. In addition, we assume the use of the TSCH (Time-Slotted, Channel Hopping) protocol (a recent addition to the IEEE 802.15.4 standard) and adopt the electric charge measurements provided in [31] for the GINA platform [32] ($Q_{tx} = 69.6$ μC, $Q_{rx} = 72.1$ μC, $V = 3.3$ V). We consider five times higher consumption for sub-GHz IEEE 802.15.4 to account for the lower transmission rate. All reported EK-128 values are in mJ. It is highlighted that the energy required to generate a key is platform-dependent, and thus the reported EK-128 values are indicative. The SBP metric, on the other hand, is more suitable for platform-agnostic comparisons.

Lastly, we confirm the randomness of the generated keys with the NIST statistical test suite [33].

5.2. Indoor applications at 2.4 GHz

In this section, SKYGlow is evaluated in an indoor environment, as illustrated in Fig. 2. This is a common environment for indoor monitoring applications. A summary of the results of the indoor experiments can be seen in Table 2.

Alice, Bob and Eve remain stationary: During office hours (09:00–17:00), the environment is busy. Human activity makes the channel dynamic and, thus, increases the entropy of the measurements [11]. Fig. 7 shows that there is strong correlation between the RSS logged by Bob and Alice, whilst there is significantly worse correlation between the RSS logged by Bob and Eve.

We examine two types of links, with and without LoS. In the case of SKYGlow.a with LoS the key agreement rate is 0.835, whilst 77.66 messages are required on average to generate a key of 128 bits. This translates to an SBP of 1.64 and an EK-128 of 36.3 mJ on average. SKYGlow.c yields higher key agreement probability (0.902) at the cost of efficiency (0.89 secret bits per packet). Indeed an average of 142.9 packets are required to generate a key and the EK-128 is 66.8 mJ on average. In the non Line of Sight (nLoS) case, SKYGlow performs better, because of the fact that the channel is more dynamic, as it relies on multipath components. More specifically, SKYGlow.a yields a higher KAR (0.897) and requires 72.34 messages to generate a key is 72.34. In turn, SBP is 1.77 and EK-128 is 33.8 mJ on average.

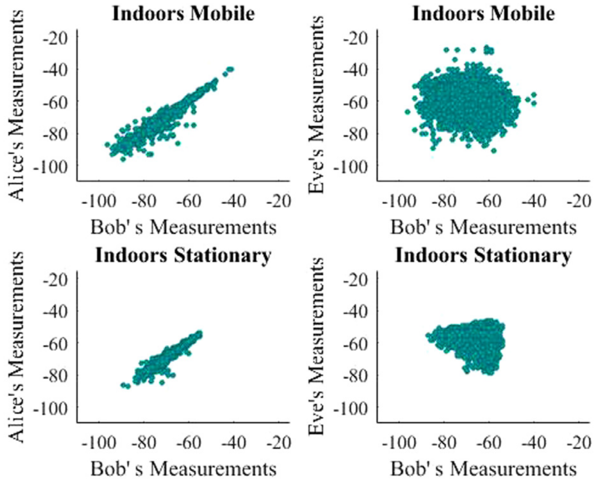


Fig. 7. Scatter plots illustrating the correlation of measurements for both the stationary with LoS, and mobile indoor scenarios.

Alice is mobile, Bob and Eve remain stationary: Next, we examine the scenario of a link between a stationary and a mobile terminal, *e.g.* off-body communications. This scenario was executed twice, yielding similar results. As expected, mobility results to the highest performance due to the channel dynamics. SKYGlow.a yields a key generation probability of 0.844, requiring 67.36 packets on average to generate a 128-bit key. The rate is 1.90 bits per packet and the EK-128 is 39.5 mJ on average. When employing SKYGlow.c the KAR rises to 0.938, needing 137.46 messages on average to generate the key, meaning that the SBP drops to 0.93. The reader can refer to Fig. 7, where the correlation of the RSS for the legitimate and the wiretap channel is illustrated.

5.3. Outdoor applications at 868 MHz

Although indoor scenarios are a large part of IoT applications, it is equally important to examine how any secret key generation scheme works in an outdoor setting. The 868 MHz band has been used extensively for these type of applications thanks to the large transmission range (some communication protocols like Sigfox's claim up to 40 km where LoS is available [34]). Furthermore, the energy-conscious design of SKYGlow complements well the energy-saving features of Low Throughput Networks as defined by ETSI's specifications. We proceed then to evaluate SKYGlow's performance on the 868 MHz band in a variety of topologies. A summary of the results of the outdoor experiments can seen in Table 3.

Alice, Bob and Eve remain stationary: Our first examined outdoor scenario assumes that both Alice and Bob, as well as Eve remain stationary. Alice and Bob are positioned in opposing sides of a busy road with constant traffic and pedestrians. In such a situation, as we can see in Fig. 8, there is significant correlation when there seems to be LoS, which degrades as the power of the received messages get weaker. Lack of correlation can limit the performance of SKYGlow.a, in which case it is advisable to switch to SKYGlow.c. Similarly to the indoor experiments, the correlation between Bob and Alice is very strong (0.9726), yet significantly weaker between Eve and Bob (0.7353).

Alice is mobile, Bob and Eve remain stationary: Next, we examine the scenario where one of the transceivers is mobile, while the other remains stationary. We can see in Fig. 3 the path that the mobile terminal followed. We present two different cases for the mobile scenario, the first having Alice moving at a slow pace, while in the second Alice was moving at a fast pace. As we can

Please cite this article as: G. Margelis et al., Efficient DCT-based secret key generation for the Internet of Things, Ad Hoc Networks (2018), https://doi.org/10.1016/j.adhoc.2018.08.014

Table 3
Performance metrics for outdoor scenarios at 868 MHz.

	SKYGlow.a			SKYGlow.c		
	Mobile 1	Mobile 2	Stationary	Mobile 1	Mobile 2	Stationary
BER (A)	0.078	0.109	0.004	0.028	0.015	0.001
BER (E)	0.488	0.491	0.495	0.492	0.485	0.497
KAR	0.750	0.687	0.892	0.866	0.886	0.909
KLR	0.002	0.003	0.001	0.001	0.003	0.001
SBP	1.48	1.36	1.76	0.86	0.88	0.894
Entropy	0.997	0.997	0.997	0.997	0.998	0.997
E[N]	86.33	94.15	72.74	148.80	145.46	143.22
E[M]	0.173	0.283	0.073	0.149	0.438	0.143
EK-128	201.8	220.1	170.1	347.9	340.1	334.9

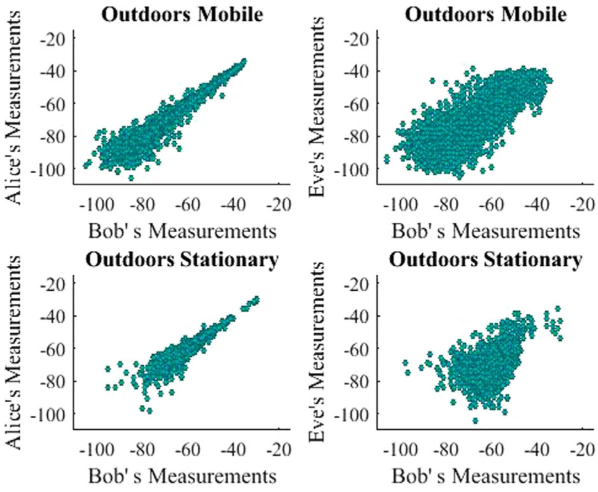


Fig. 8. Scatterplots illustrating the correlation of measurements for both the stationary and mobile outdoor scenarios.

Table 4
The results of testing 10,000 SKYGlow keys with the NIST test suite [33]. The pass rate is approximately 9870/10000.

Statistical test	Success proportion
“Frequency (Monobits) Test”	9972/10,000
“Frequency Test Within a Block”	9972/10,000
“Cumulative Sums (Cusum) Test (early stages)”	9976/10,000
“Cumulative Sums (Cusum) Test (late stages)”	9975/10,000
“Approximate Entropy Test”	10,000/10,000
“Serial Test #1”	9972/10,000
“Serial Test #2”	9972/10,000

Table 5
Performance comparison at 2.4 GHz.

	SBP	EK-128	Entropy
Ali et al. [21]	0.33–0.37	161.8–181.4	0.9979
Patwari et al. [18]	0.05–0.44	136–1197.1	0.9590
Li et al. [41]	0.65–0.75	79.8–92.1	0.993
SKYGlow.a	1.36–1.9	33.8–39.5	0.9971
SKYGlow.c	0.86–0.93	64.3–66.8	0.9979

observe in Fig. 8, there is now a greater variation of RSS values in the legitimate channel, although correlation reduces as the RSS weakens in similar fashion with the stationary scenario. Again, the correlation between Bob and Alice is very strong (0.9728), whilst the correlation of the RSS measurements logged by Eve and Bob is significantly weaker (0.8110).

5.4. NIST statistical tests

A vital requirement for a key generation algorithm is to generate keys that are *random*. We evaluate the randomness of the keys that we generate with SKYGlow using the NIST statistical test suite [33]: a suite that is regularly used to measure the outputs of random or pseudo-random number generators that may he used in many cryptographic applications, such as the generation of key material.

SKYGlow successfully passed the *“Frequency (Monobits) Test”* and the *“Frequency Test Within a Block”*, both versions of the *“Cumulative Sums Test”*, the *“Approximate Entropy Test”* and both versions of the *“Serial Test”*. Maurer’s *“Universal Test”*, the *“Discrete Fourier Transform, Random Excursions Test”*, and its variant, *“Binary Matrix Rank, Linear Complexity Test”* were not examined as they are designed to test sequences significantly larger than the ones generated by SKYGlow. The results of testing 10,000 keys generated by SKYGlow can be seen in Table 4.

5.5. Discussion on experimental results

First of all, the results suggest that SKYGlow is characterized by a very high key agreement probability. Indeed, it is practically guaranteed that a key will be generated after exchanging 195 messages, as the probability of three consecutive failed attempts is less than 1%.

Secondly, the results suggest that the entropy and correlation of the RSS observations control the performance of SKYGlow. When the environment is very static and the entropy of the observations is very low, the RSS measurements are characterized by uncorrelated randomness (due to, *e.g.* thermal noise) that makes the performance drop. Fig. 9 shows the effects of the entropy and correlation of the RSS observations on the performance of SKYGlow.a. The figure suggests that more than 1.2 bits of entropy and more than 0.7 correlation are needed for agreeing a key in less than 200 packet transmissions.

Our measurements in real-world environments suggest that human activity makes the environment sufficiently dynamic to satisfy these requirements. Indeed, the indoor measurements suggest that during office hours, the correlation is generally greater than 0.8, whilst the entropy is greater than 1.5, in both stationary and mobile links, as shown in Fig. 10. Yet, these requirements are not satisfied during non-working hours (00:00-06:00) where there is no human activity in the environment. In such cases, generating secret keys with SKYGlow, or any other secret key generation scheme for that matter, becomes inefficient. A possible solution to this could be to create keys during high-entropy periods that are then stored and used in low-entropy periods. However, storing keys locally for future use, opens up a new host of potential vulnerabilities. Further exploration of this issue is considered out of the scope of this article.

5.6. Comparison with other schemes

Table 5 compares the performance of SKYGlow against three schemes in the literature that are also suitable for resource-

Please cite this article as: G. Margelis et al., Efficient DCT-based secret key generation for the Internet of Things, Ad Hoc Networks (2018), https://doi.org/10.1016/j.adhoc.2018.08.014

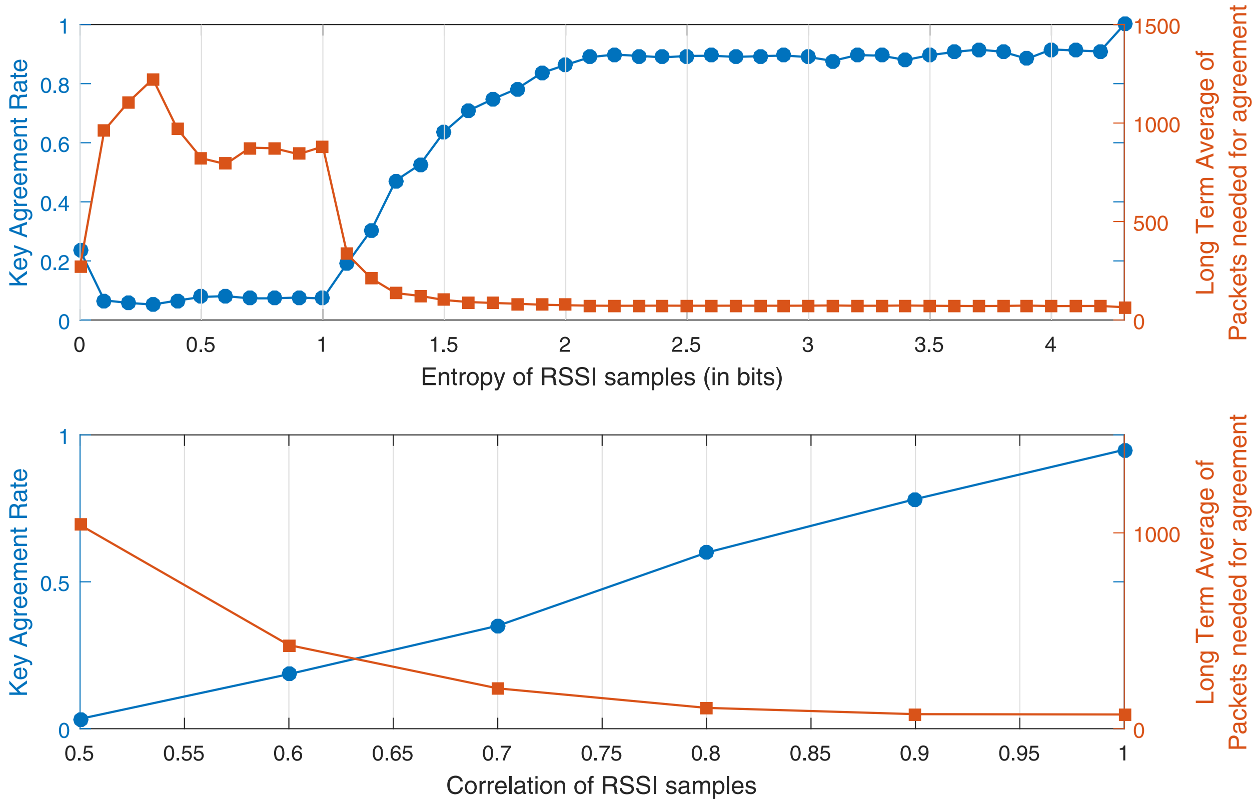


Fig. 9. The effect of entropy (top) and correlation (bottom) on the key agreement rate (blue) and E[N] (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

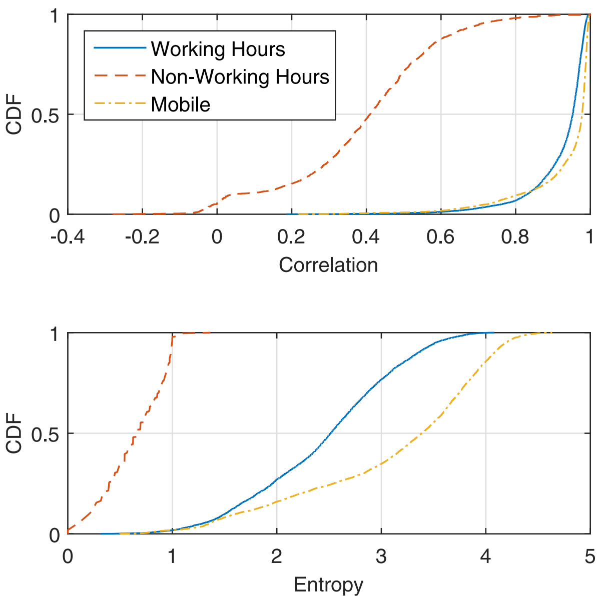


Fig. 10. Empirical CDF of the correlation (top) and the entropy (bottom) for mobile links, and stationary links during working hours and non-working hours.

constrained IoT platforms. Interested in energy efficiency rather than speed, we do not compare the key generation schemes in terms of bits per second (bps), as typically seen in the literature. Instead, we use the SBP and EK-128 metrics. As a result, we omit [35] from the comparison, as it is not possible to derive these metrics from the published paper. We note however that SKYGlow outperforms [35] by one order of magnitude (1.64 bps against 0.195 bps). Furthermore, we limit the scope of the comparison to battery-powered IoT platforms with a single antenna that are based on low-power wireless standards, such as IEEE 802.15.4. Schemes like [36–39] are unsuitable for such IoT platforms as they employ multiple antennas or frequency diversity. We also limit our scope to schemes that support point-to-point links. Therefore, we omit [40] as it assumes topologies with pre-authenticated relays.

It is also highlighted that an important difference between SKYGlow and [41] (included in the comparison) is that the latter assumes full duplex communication, and is evaluated on devices that have the ability to transmit and receive concurrently in different frequency channels (frequency-division duplex). SKYGlow, on the other hand, does not require full duplex and is, therefore, applicable to platforms that do not support this functionality.

Overall, Table 5 demonstrates that, similarly to the related works, the keys that are generated by SKYGlow have sufficiently high entropy. The important difference lies on the fact that key generation is more energy-efficient, *i.e.* Alice and Bob need to exchange fewer messages in order to establish a shared key.

Although SKYGlow.a outperforms other proposed schemes in the literature, there is still significant space for improvement as at its best it can only generate 1.9 bits per message exchange, while the predicted maximum for the same scenario, based on the MI, is approximately 3 bits per message exchange.

6. Conclusions

This article first identifies the practical upper bounds of the key capacity between two off-the-shelf IEEE 802.15.4-based IoT devices, in real-world experiments that consider a variety of practical settings, such as indoor links, outdoor links, and mobile links. The article then presents SKYGlow: an energy-efficient secret key generation algorithm, suitable for resource-constrained IoT devices. The proposed scheme introduces a DCT stage between the sampling and quantization stage, resulting in bit sequences that are characterized high entropy, thus making the privacy amplification stage

Please cite this article as: G. Margelis et al., Efficient DCT-based secret key generation for the Internet of Things, Ad Hoc Networks (2018), https://doi.org/10.1016/j.adhoc.2018.08.014

unnecessary. Moreover, by operating on the frequency domain, the process is tunable, allowing the protocol to efficiently discard uncorrelated high-frequency components. SKYGlow is able to generate secret keys of high entropy with few packet exchanges (*i.e.*, up to 1.9 and 1.64 bits per packet in mobile and stationary scenarios respectively), and thus constitutes an efficient solution for encryption key generation by energy-constrained IoT devices.

Acknowledgement

This work was supported by EPSRC through grants EP/I028153/1 and EP/K031910/1 (IRC-SPHERE). Additional research data supporting this publication are available in https://github.com/irc-sphere/skyglow.

References

[1] X. Fafoutis, L. Marchegiani, G.Z. Papadopoulos, R. Piechocki, T. Tryfonas, G. Oikonomou, Privacy leakage of physical activity levels in wireless embedded wearable systems, IEEE Signal Process. Lett. 24 (2) (2017) 136–140, doi:10.1109/LSP.2016.2642300.

[2] W. Diffie, M.E. Hellman, New directions in cryptography, Inf. Theory IEEE Trans. 22 (6) (1976) 644–654.

[3] A.F. Skarmeta, J.L. Hernandez-Ramos, M. Moreno, A decentralized approach for security and privacy challenges in the internet of things, in: Internet of Things (WF-IoT), 2014 IEEE World Forum on, IEEE, 2014, pp. 67–72.

[4] M. Abomhara, G.M. Koien, Security and privacy in the internet of things: current status and open issues, in: Privacy and Security in Mobile Systems (PRISMS), 2014 International Conference on, IEEE, 2014, pp. 1–8.

[5] C. Ye, P. Narayan, Secret key and private key constructions for simple multiterminal source models, Inf. Theory IEEE Trans. 58 (2) (2012) 639–651.

[6] J. Muramatsu, K. Yoshimura, P. Davis, A. Uchida, T. Harayama, Secret-key distribution based on bounded observability, Proc. IEEE 103 (10) (2015) 1762–1780.

[7] A.A. Hassan, W.E. Stark, J.E. Hershey, S. Chennakeshu, Cryptographic key agreement for mobile radio, Digit. Signal Process. 6 (4) (1996) 207–212.

[8] R. Wilson, D. Tse, R.A. Scholtz, Channel identification: secret sharing using reciprocity in ultrawideband channels, Inf. Forensics Secur. IEEE Trans. 2 (3) (2007) 364–375.

[9] R. Ahlswede, I. Csiszár, Common randomness in information theory and cryptography. part I: secret sharing, IEEE Trans. Inf. Theory 39 (4) (1993).

[10] R. Ahlswede, I. Csiszár, Common randomness in information theory and cryptography. ii. cr capacity, Inf. Theory IEEE Trans. 44 (1) (1998) 225–240.

[11] G. Margelis, X. Fafoutis, G. Oikonomou, R.J. Piechocki, T. Tryfonas, P. Thomas, Practical limits of the secret key-capacity for IoT physical layer security, 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT) (WF-IoT 2016), Reston, USA, 2016.

[12] G. Margelis, X. Fafoutis, G. Oikonomou, R.J. Piechocki, T. Tryfonas, P. Thomas, Physical layer secret-key generation with discreet cosine transform for the internet of things, in: IEEE International Conference in Communication (ICC), Paris, France, 2017.

[13] S. Mathur, W. Trappe, N. Mandayam, C. Ye, A. Reznik, Radio-telepathy: extracting a secret key from an unauthenticated wireless channel, in: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, ACM, 2008, pp. 128–139.

[14] M.G. Madiseh, M.L. McGuire, S.W. Neville, A.A.B. Shirazi, Secret key extraction in ultra wideband channels for unsynchronized radios, in: Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual, IEEE, 2008, pp. 88–95.

[15] U.M. Maurer, S. Wolf, Unconditionally secure key agreement and the intrinsic conditional information, Inf. Theory IEEE Trans. 45 (2) (1999) 499–514.

[16] S.N. Premnath, S. Jana, J. Croft, P.L. Gowda, M. Clark, S.K. Kasera, N. Patwari, S.V. Krishnamurthy, Secret key extraction from wireless signal strength in real environments, IEEE Trans. Mob. Comput. 12 (5) (2013) 917–930.

[17] S.N. Premnath, J. Croft, N. Patwari, S.K. Kasera, Efficient high-rate secret key extraction in wireless sensor networks using collaboration, ACM Trans. Sens. Netw. (TOSN) 11 (1) (2014) 2.

[18] N. Patwari, J. Croft, S. Jana, S.K. Kasera, High-rate uncorrelated bit extraction for shared secret key generation from channel measurements, IEEE Trans. Mob. Comput. 9 (1) (2010) 17–30.

[19] A.J. Pierrot, R.A. Chou, M.R. Bloch, Experimental aspects of secret key generation in indoor wireless environments, in: 2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), IEEE, 2013, pp. 669–673.

[20] M. Bloch, J. Barros, M.R. Rodrigues, S.W. McLaughlin, Wireless information-theoretic security, IEEE Trans. Inf. Theory 54 (6) (2008) 2515–2534.

[21] S.T. Ali, V. Sivaraman, D. Ostry, Zero reconciliation secret key generation for body-worn health monitoring devices, in: Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks, ACM, 2012, pp. 39–50.

[22] G.D. Durgin, T.S. Rappaport, Theory of multipath shape factors for small-scale fading wireless channels, Antennas Propag., IEEE Trans. 48 (5) (2000) 682–693.

[23] W. Trappe, The challenges facing physical layer security, IEEE Commun. Mag. 53 (6) (2015) 16–20, doi:10.1109/MCOM.2015.7120011.

[24] Texas Instruments, CC2650 simple link multistandard wireless MCU, 2015, http://www.ti.com/lit/ds/symlink/cc2650.pdf.

[25] Texas Instruments, CC1310 simple link ultra-low-power Sub-1 GHz wireless MCU, 2016, http://www.ti.com/lit/ds/symlink/cc1310.pdf.

[26] S. Eberz, M. Strohmeier, M. Wilhelm, I. Martinovic, A practical man-in-the-middle attack on signal-based key generation protocols, in: European Symposium on Research in Computer Security, Springer, 2012, pp. 235–252.

[27] G. Brassard, L. Salvail, Secret-key Reconciliation by Public Discussion, Springer-Verlag, 1994, pp. 410–423.

[28] J. Martinez-Mateo, C. Pacher, M. Peev, A. Ciurana, V. Martin, Demystifying the information reconciliation protocol cascade, arXiv:1407.3257 (2014).

[29] A. Elsts, R. McConville, X. Fafoutis, N. Twomey, R. Piechocki, R. Santos-Rodriguez, I. Craddock, On-board feature extraction from acceleration data for activity recognition, in: International Conference on Embedded Wireless Systems and Networks (EWSN), ACM, 2018.

[30] A. Kinane, V. Muresan, N. O'Connor, N. Murphy, S. Marlow, Energy-efficient hardware architecture for variable n-point 1d dct, in: E. Macii, V. Paliouras, O. Koufopavlou (Eds.), Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 780–788.

[31] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, K.S.J. Pister, A realistic energy consumption model for TSCH networks, IEEE Sens. J. 14 (2) (2014) 482–489, doi:10.1109/JSEN.2013.2285411.

[32] A.M. Mehta, K.S.J. Pister, WARPWING: A complete open source control platform for miniature robots, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 5169–5174.

[33] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, Technical Report, DTIC Document, 2001.

[34] G. Margelis, D. Kaleshi, R.J. Piechocki, P. Thomas, Low throughput networks for the IoT: lessons learned from industrial implementations, 2015 IEEE World Forum on Internet of Things (WF-IoT) (WF-IoT 2015), Milano, Italy, 2015.

[35] S.A. Salehi, M. Razzaque, I. Tomeo-Reyes, N. Hussain, V. Kaviani, Efficient high-rate key management technique for wireless body area networks, in: Communications (APCC), 2016 22nd Asia-Pacific Conference on, IEEE, 2016, pp. 529–534.

[36] G. Revadigar, C. Javali, W. Hu, S. Jha, Dlink: Dual link based radio frequency fingerprinting for wearable devices, in: Local Computer Networks (LCN), 2015 IEEE 40th Conference on, IEEE, 2015, pp. 329–337.

[37] L. Yao, S.T. Ali, V. Sivaraman, D. Ostry, Decorrelating secret bit extraction via channel hopping in body area networks, in: Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on, IEEE, 2012, pp. 1454–1459.

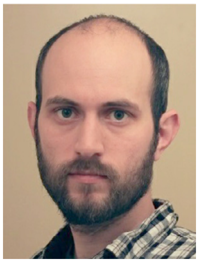
[38] G. Revadigar, C. Javali, H.J. Asghar, K.B. Rasmussen, S. Jha, Mobility independent secret key generation for wearable health-care devices, in: Proceedings of the 10th EAI International Conference on Body Area Networks, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015, pp. 294–300.

[39] J. Choi, Secret key transmission for OFDM based machine type communications, J. Commun. Netw. 19 (4) (2017) 363–370, doi:10.1109/JCN.2017.000060.

[40] S. Xiao, Y. Guo, K. Huang, L. Jin, High-rate secret key generation aided by multiple relays for internet of things, Electron. Lett. 53 (17) (2017) 1198–1200, doi:10.1049/el.2017.2346.

[41] Z. Li, Q. Pei, I. Markwood, Y. Liu, H. Zhu, Secret key establishment via rss trajectory matching between wearable devices, IEEE Trans. Inf. Forensics Secur. 13 (3) (2018) 802–817, doi:10.1109/TIFS.2017.2768020.

George Margelis received a B.Sc. in Physics and an M.Sc. in Microelectronics and Electronic Physics from the Aristotle University of Thessaloniki (Greece), in 2006 and 2008 respectively, followed by a Ph.D. related to the security aspects of the Internet Of Things from the University of Bristol (UK) in 2017. He is currently working in the patent law sector, specializing in electronics, communications and IoT intellectual property.



Xenofon Fafoutis received a PhD degree in Embedded Systems Engineering from the Technical University of Denmark in 2014; an MSc degree in Computer Science from the University of Crete in 2010; and a BSc in Informatics and Telecommunications from the University of Athens in 2007. From 2014 to 2018, he held various researcher positions at the University of Bristol. Since 2018 he is an Assistant Professor with the Technical University of Denmark (DTU). His research interests primarily lie in Networked Embedded Systems as an enabling technology for Digital Health, Smart Cities, Industry 4.0, and the Internet of Things (IoT).

Please cite this article as: G. Margelis et al., Efficient DCT-based secret key generation for the Internet of Things, Ad Hoc Networks (2018), https://doi.org/10.1016/j.adhoc.2018.08.014

George Oikonomou received the M.Sc. and Ph.D. degrees in computer science from the Athens University of Economics and Business, Athens, Greece, in 2002 and 2009, respectively. He is currently a Lecturer with Electrical and Electronic Engineering, University of Bristol, Bristol, U.K. His current research focuses on low-power networking and security for severely constrained wireless embedded devices. He is also interested in digital forensics for emerging technologies, such as smartphones, wireless sensor networks and the Internet of Things. George is one of the maintainers of the Contiki open source embedded operating system for the Internet of Things.

Robert Piechocki received an M.Sc. degree from Technical University of Wroclaw (Poland) in 1997 and a Ph.D. degree from the University of Bristol in 2002. Robert is currently a reader in Advanced Wireless Access and a member of Communications Systems and Networks group. His research interests span the areas of Statistical Signal Processing, Information and Communication Theory, Wireless Networking, Body and ad-hoc networks, Ultra Low Power Communications and Vehicular Communications. He has published over 100 papers in international journals and conferences and holds 13 patents in these areas.

Theo Tryfonas is a Reader in Smart Cities with a background in cyber security, systems engineering and software development. His research expertise includes secure and resilient operation of Internet of Things (IoT) applications, privacy in mobile computing, energy efficient deployments of wireless sensor networks and open data architectures for smart buildings/infrastructure. He received a B.Sc. in Computer Science from the University of Crete, Greece in 1996, as well as an M.Sc. in information systems and a Ph.D. in informatics from Athens University of Economics and Business, Athens, Greece in 1998 and 2003, respectively.

Paul Thomas holds a B.Sc in computer science and in 1996 received a Ph.D in electrical engineering from the University of Bristol. He is currently a visiting Research Fellow with the Faculty of Engineering, University of Bristol, Bristol UK. He has extensive industrial experience in power electronics, high speed computing and wireless networks. His current research interests are new security models and techniques that can be applied to low power wireless sensors prevalent in Internet of Things networks.