Traffic Analysis On 6LoWPAN

Yan Yan

March 15, 2017

1 Introduction

Traffic Analysis is a well studied technique that breaches data confidentiality over Internet which exploits the side channel information, such as packet length and timing, to reveal contents in the encrypted traffic. As a matter of fact, Traffic Analysis techniques is often associated with malicious Internet behaviour such as privacy violation and mass surveillance.

On the other hand, the development of Internet of Things (IoT) has linked ever more private data to the network, with radio being the most extensively used media for data transmission. These factors combined made Traffic Analysis posing even greater threat in the IoT context, as data are made widely open yet its content are juicier than ever.

In this paper, we revised the applicability of Traffic Analysis in IoT applications by demonstrating simple examples capturing the concept. We made extensive use of Contiki OS [1] in our experiments to build demo applications. A 6LoWPAN [2] network is built on two models of typical devices, namely TelosB [3] and CC2538 [4], as well as a network constructed by simulated Wismote [5].

The paper is structured as follows. We first review the related literatures in Section 2 and briefly report a security flaw in Contiki source code in Section 3. We then present the first example that extracts ICMP[1] [6] messages in Section 4, followed by another example that reveals hardware information in Section 5. Section 6 proposes a new type of side channel attack, we named PingProbe, that fingerprints the application running on an IoT node. Finally we conclude the paper in Section 7.

2 Related Work

Traffic Analysis is a well studied subject in the Internet security and privacy community. [7] summarised several scenario of side channel attacks against web applications, followed by [8] which proposes the use of Mutual Information to pinpoint the potential points of information leakage. [9], [10] and [11] described

[1] Specifically, the term ICMP we used in this paper refers to ICMPv6.

attacks against encrypted text, voice and video traffic respectively, while [12] instantiated an attack against Google search box. [13] and [14] studied different classifiers when adopting Machine Learning in Traffic Analysis. Different countermeasures are studied by [15], [16] and [17].

With respect to 6LoWPAN security issues, [18] summarises the known attacks in 6LoWPAN networks, including Fragmentation Attack [19], Sinkhole Attack [20], Hello Flood Attack [21], Wormhole Attack [22] and Blackhole Attack [23]. In addition, [24] reported certain problematic designs in 802.15.4 security [25].

3 802.15.4 Security in Contiki

802.15.4 security in implemented by Noncoresec [26] in the latest Contiki (release 3.0). One hard-coded network wise key is the only keying model it supports in the current release. Due to the absence of IPSec [27], noncoresec is effectively the only applicable approach for securing part of the network metadata, such as IP headers and TCP/UDP headers.

However, examining the code, we realised two issues exist in the current implementation:

- **Nonce Reusing**
  Figure 1 illustrates the construction of nonce in 802.15.4 security with AES-128 in CTR and CCM mode. According to the specification [25], for a specific source, the Frame Counter is the only variable field among different packets. Noncoresec implemented Frame Counter as a static variable initialised to 0 upon each reboot. Such implementation eventually leads to the nonce being reused upon each reset of the device, which in many cases could cause severe leakage to the plaintext.

1 (bytes)	8	4	1	2
Flags	Source Address	Frame Counter	Security Level	Block Counter

Figure 1: Nonce Construction in 802.15.4 Security

- **Packet Lost by Anti-replay**
  Even though the incompatibility between Anti-replay and network key has been pointed out by [24], the noncoresec implementation has mitigated the issue by using an extra data structure recording the last Frame Counter from each source address. However, this approach also induces a problem that packets sent by a rebooted device will be labelled as replays and thus dropped by the receiver.

Although nonce reusing could be mitigated by initialising Frame Counter to a random value on each reboot, or recording the latest value on a permanent media such as flash memory, yet the 4 byte space can hardly be considered

cryptographically secure. Therefore a full solution may not be achieved without updating the nonce construction specified by the standard [25].

The anti-replay issue is closely related to key management which remains an open question in the subject; thus would either be easily to solved without modifying the standard. As a compromising solution, we simply recommend to disable the anti-replay feature and leave it to upper layer protocols such as TCP and CoAP.

4 Extract ICMP Messages

ICMP messages are used for network maintenance. Their typical usage include update routing information, network debugging, etc. Further details of ICMP messages are specified in [6].

As of the latest Contiki release 3.0, neither IPSec [27] nor Secure ICMP Messages, i.e. ICMP Messages with encryption and authentication, are implemented; thus leaving 802.15.4 Security the only option for encrypting ICMP messages.

Due to the nature of OSI layered model [28], ICMP messages are independent to upper layer applications; thus they hardly contain any information of the application data. Yet from a security perspective, leakage of ICMP messages may still endanger the network. For example, the identity of root node could guide a Denial of Service attack to amplify its damage, and being able filter some specific ICMP messages could be exploited to conduct other attacks such as the Wormhole attacks [22], etc.

We simulated a 6LoWPAN network constituted of multiple Wismote [5] nodes running Contiki broadcast and unicast examples, with 802.15.4 Security set to the highest level (CCM* with 128 bit MIC). The ICMP messages generated in our simulation includes:

- **DAG Information Object (DIO)**
  DIO contains the 6LoWPAN global information. It could be periodically broadcasted for network maintenance, or unicasted to a new joining node as a reply to DIS (see below).
- **DAG Information Solicitation (DIS)**
  DIS is sent by a new started node to probe any existing 6LoWPANs. A DIO would be replied if the DIS is received by any neighbour nodes.
- **Destination Advertisement Object (DAO)**
  DAO is sent by a child node to its precedents[2] to propagate its routing information.
- **Neighbour Solicitation (NS) and Neighbour Advertisement (NA)**
  NS and NA are the ARP replacement in IPv6, where NS queries a translation and NA answers one. In addition, they are also used for local link validity check.

[2]The 6LoWPAN DODAG topology is defined in [29].

- **Echo Request and Echo Response (PING)**
  Echo Request and Echo Response are well known as the PING packets. They are mostly used for diagnostic purposes, such as connectivity test or Round Trip Time (RTT) estimation. Echo Request may contain arbitrary user defined data and Echo Response simply echoes its corresponding request.

DIO/DIS/DAO and NS/NA are defined by [29] and [30] respectively. PING is defined by [31].

Our simulation shows that even though the exact content of these messages are hidden under the encryption provided by 802.15.4 Security, some of them are still distinguishable simply by their packet size and type of MAC destination. We summarise these features in Table 1 where x is the size of user defined data in PING packets.

Considering the fact that UDP is the relatively preferable Transportation Layer protocol in IoT applications than TCP, we also attached the packet features of UDP multicast and unicast at the bottom of Table 1.

There are two points may worth be noticed for Table 1:

- Both DIO and NS can be sent in either broadcast or unicast. The broadcasted DIO is smaller than unicasted as it uses an abbreviated IPv6 multicast address "ff02::1a". Broadcasted NS uses another multicast address "ff02::1:ff00:0" which has the same length as unicast. However, both of them are mapped to the same "0xffff" Link Layer broadcast address in 802.15.4 MAC Header.

- The size of PING may vary due to different user defined data. According to [2], any packet less than the 802.15.4 MTU, i.e. 127 bytes, should not be fragmented; however, we realised that Contiki fragments PING larger than 107 bytes. We have not identified the cause but we consider this might be an implementation bug.

	Packet Size (bytes)	Type of MAC Destination
DIS	85	broadcast
DIO	118/123	broadcast/unicast
DAO	97	unicast
NS	87	broadcast/unicast
NA	87	unicast
PING	$101 + x$	unicast
UDP Multicast	$85 + x$	broadcast
UDP Unicast	$107 + x$	unicast

Table 1: ICMPv6 Packets in Simulated 6LoWPAN with noncoresec, where x is size of user defined data

Observing Table 1, we realised 5 among the ICMP messages can be distinguished from all other packets, as summarised in Table 2.

	(Size, MAC Destination)
DIS	(85, broadcast)
DAO	(97, unicast)
NS (broadcast)	(87, broadcast)
NS (unicast)	(87, unicast)
NA	(87, unicast)

Table 2: Distinctive ICMP Packets

Other packets in Table 1 may still be distinguishable in the encrypted traffic, unless the upper application happens to generate packets with the exact features.

Packet fragmentation may be a major factor that induces false result when applying this method. However it is likely to be avoided by most applications.

5 Distinguish Hardware

IoT applications utilise a great variation of devices, with each having different processing power. Consequently, the time required for a specific operation could be exploited as an indicator to the hardware.

The time required to process a PING Request could therefore be exploited to perform such hardware distinguishing attack. Although the same attack could be conducted on various packets, PING is specifically practical and ideal due to:

1. PING is processed in a highly predictable manner where nearly no computation is required; thus minimises the noise induced by data dependency.

2. Support to PING is mandatory according to the standard [6], making the attack universally applicable.

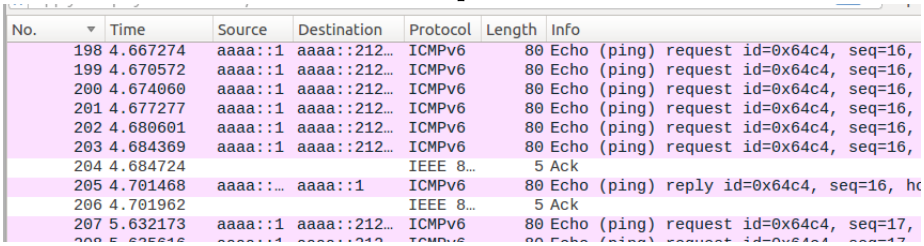
For our experiments, the PING Requests are initiated from a Linux machine connected into the 6LoWPAN through a border router. The "-s 0" option is specified for the 'ping6' command in order to avoid exceeding the MTU of 6LoWPAN (127 bytes).

5.1 Computing PING Response Time

For energy preservation, Contiki adopts a Radio Duty Cycle (RDC) protocol called Contiki MAC [32]. On receiver's side, Contiki MAC switches off the radio for most of the times and periodically wakes up for a short period for signal detection. If a signal is detected during the awaken period, the radio is kept on to receive a packet, followed by an acknowledgement sent to inform the sender. On sender's side, a packet is repeatedly transmitted until timed out or an acknowledgement is received.

Therefore when ContikiMAC is enabled by default, duplicated packets may be observed in the captured traffic. As an approximation to the PING Request processing time, we define PING Response Interval (PRI) as the time elapsed between the last PING Request and the first PING Response.

Example 1.



No.	Time	Source	Destination	Protocol	Length	Info
198	4.667274	aaaa::1	aaaa::212…	ICMPv6	80	Echo (ping) request id=0x64c4, seq=16,
199	4.670572	aaaa::1	aaaa::212…	ICMPv6	80	Echo (ping) request id=0x64c4, seq=16,
200	4.674060	aaaa::1	aaaa::212…	ICMPv6	80	Echo (ping) request id=0x64c4, seq=16,
201	4.677277	aaaa::1	aaaa::212…	ICMPv6	80	Echo (ping) request id=0x64c4, seq=16,
202	4.680601	aaaa::1	aaaa::212…	ICMPv6	80	Echo (ping) request id=0x64c4, seq=16,
203	4.684369	aaaa::1	aaaa::212…	ICMPv6	80	Echo (ping) request id=0x64c4, seq=16,
204	4.684724			IEEE 8…	5	Ack
205	4.701468	aaaa::…	aaaa::1	ICMPv6	80	Echo (ping) reply id=0x64c4, seq=16, hc
206	4.701962			IEEE 8…	5	Ack
207	5.632173	aaaa::1	aaaa::212…	ICMPv6	80	Echo (ping) request id=0x64c4, seq=17,

Figure 2: PRI Example

Figure 2 shows an example of packets captured by Wireshark [33]. Duplicated PING Requests, No.198 to No.203, are replied by a single PING Response, No.205, matched by their *seq* field (16). The PRI is therefore computed as the difference in time between No.203 (last PING Request) and No.205 (first PING Response) which is:

$$PRI = 4.701468 - 4.684369 = 0.017099(s) = 17.099(ms)$$

5.2 PRIs of TelosB and CC2538

Figure 3 shows the distributions of PRI collected on TelosB and CC2538. The bar on the right most represents the outliers in the data, with 9.43% on TelosB are $\geq$ 19ms and 2.60% on CC2538 $\geq$ 12ms respectively.

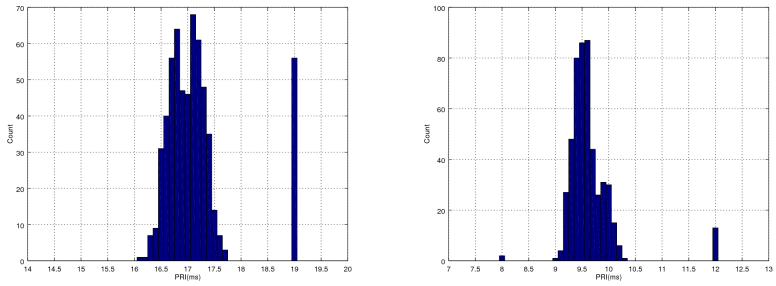
The result of Figure 3 suggests that TelosB and CC2538 can be easily distinguished by their PRIs, as they are tightly clustered around 17ms for TelosB and 9.5ms for CC2538 respectively.

With respect to the outliers, one common cause is the processor being occupied by other tasks when a PING Request is received; thus delayed the processing of PING as illustrated by Figure 4. The portion of the outlier is hence affected by the payload on the node and the frequency of PING requests. We discuss further details of the outliers in Section 6.

5.3 Factors Affecting PRI

Despite the clear distinguishability suggested by Section 5.2, there exists several practical factors the might affect the result.

802.15.4 Security [25] is a prominent factor to PRIs as it induces several cryptographic operations which are computationally heavy and hence could significantly increase the PRI. Figure 5 illustrates the PRIs on the same devices when 802.15.4 Security is enabled. Flooding PING could also make an impact



(a) TelosB PRIs (b) CC2538 PRIs

	Mean (ms)	Median (ms)
TelosB	33.82	17.03
CC2538	14.27	9.56

Figure 3: PRIs for different devices.

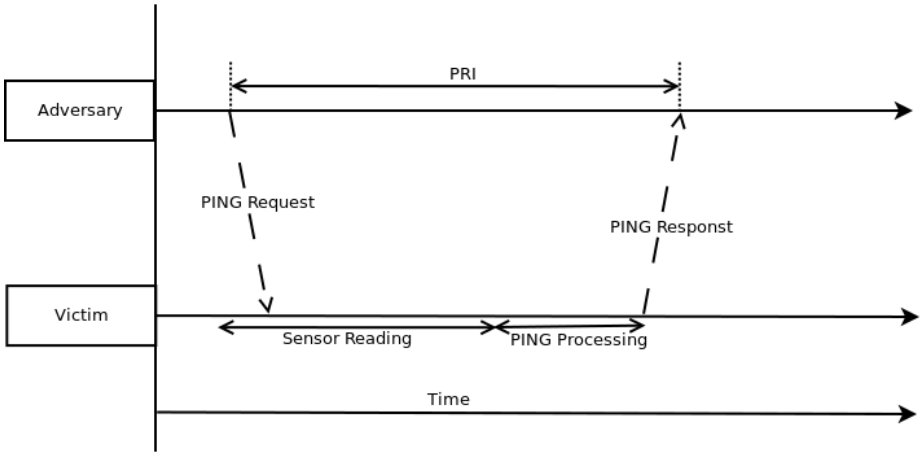


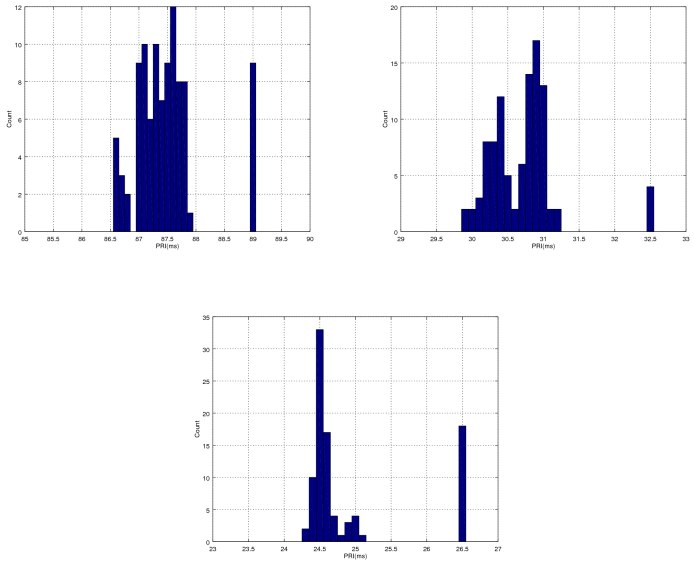
Figure 4: PRI prolonged by Sensor Reading

on measuring PRIs as it will overrun the low processing power of these resource constrained devices.

6 PingProbe: Application Fingerprint

6.1 PRI Outliers

In Section 5 we mentioned the phenomenon that processor occupied by other tasks may prolong the PRI as illustrated in Figure 4. Extracting these outliers on a CC2538 running Helloworld example using threshold of $\geq$ 11ms, we show the distribution of these prolonged PRIs in Figure 6. The histogram suggests



	Mean (ms)	Median (ms)
TelosB HW AES	37.20	30.77
TelosB SW AES	105.19	87.40
CC2538 SW AES	48.83	24.55

Figure 5: PRIs with 802.15.4 security

the distribution has a strong discrete tendency, with multiple spikes presented in the graph.

We suppose the cause of such discreteness is that the delay is indeed caused by context switching in the kernel; therefore the impact of remaining time for the current task is indeed negligible. We further suppose the existence of multiple spikes is the result of switching from different contexts take different amounts of time. However, due to practical reasons, we found it difficult to experimentally confirm these hypothesises. Nevertheless, we consider this result as a strong indication that different states of the processor would result into different value of PRI, as illustrated in Figure 7.

6.2 PingProbe Attack

The PingProbe attack is proposed to exploit the distribution of PRI outliers as an indicator of different states of the processor. Considering the fact that many IoT applications perform routine operations, e.g. reading temperature sensor periodically, the constitution of processor states can be effectively viewed as a fingerprint to the application running on the node.

To verify the effectiveness of this attack, we performed a proof of concept

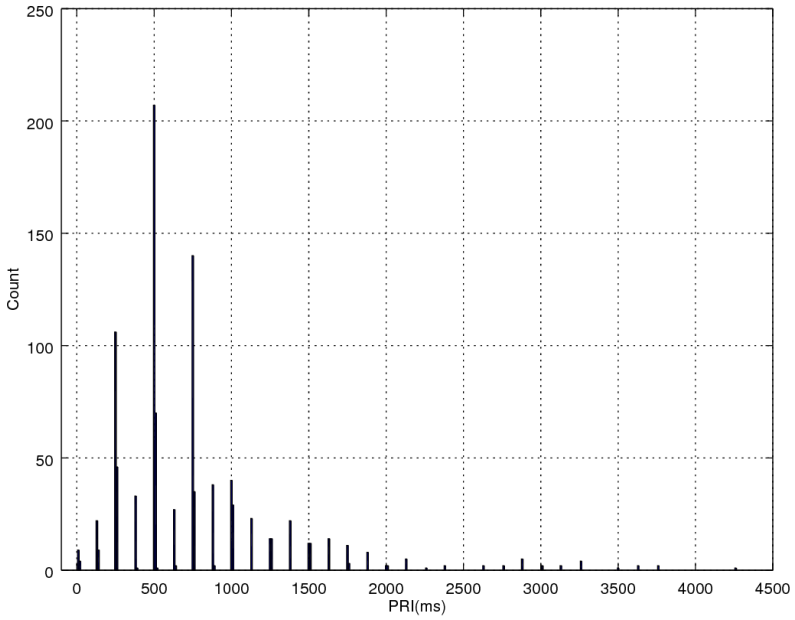


Figure 6: Histogram of PRI Outliers on CC2538 running Helloworld Example, using bins of 10ms

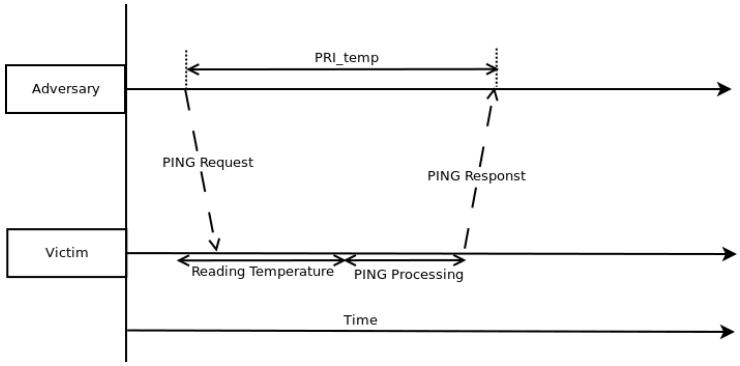
experiment.

6.2.1 Experiment Design

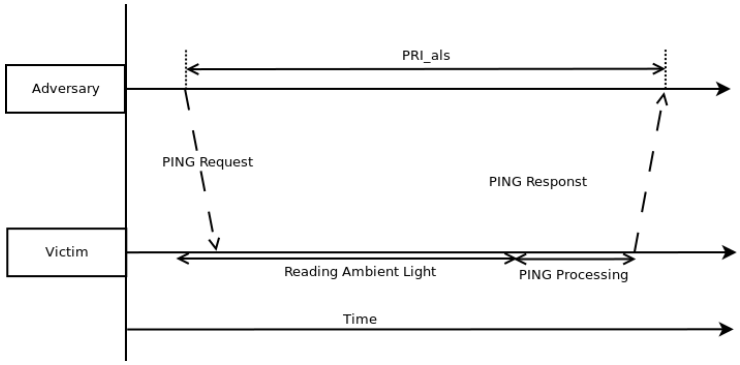
Similar to many Traffic Analysis literature, we adopted the "closed world" setting for the experiment. Formally, our experiment simulates the following scenario:

- The adversary is given the pre-knowledge of n potential candidate applications, denote as $\mathbb{A} = \{A_1, A_2, ..., A_n\}$, and a target node running an unknown application A_{x^*} where $A_{x^*} \in \mathbb{A}$. The adversary is also given the exact same hardware of target as well as the power to send PING packets to the target.
- The adversary eventually outputs x as the guess of x^*. The attack is considered effective if $P(x = x^*) > \frac{1}{n}$.

In reality, such scenario can be motivated as an adversary trying to find out which product a victim might be using among those available on the market.



(a) PRI when reading temperature sensor



(b) PRI when reading Ambient Light Sensor(ALS)

Figure 7: An example of reading temperature and ALS sensor results into different PRI.

6.2.2 Attack Description

The attack can be described into 3 stages:

Profiling To set up the attack, the adversary collects PRIs for each application in $\mathbb{A}$. We denote the profiled traces as:

$$\mathbb{T}_p = \{T_1, T_2, ..., T_n\} \tag{1}$$

where each T_i is collected by repetitively sending PING Request to the device running application A_i. Each trace T_i contains multiple PRIs, denote as:

$$T_i = \{t_{(i,1)}, t_{(i,2)}, ..., t_{(i,m_i)}\} \tag{2}$$

where m_i is the number of PRIs in T_i.

Collecting Target Trace To identify the secret application A_{x^*}, the adversary collects the PRIs on the target node running A_{x^*}. We denote the

target trace as:

$$T_{x^*} = \{t_{(x^*,1)}, t_{(x^*,2)}, ..., t_{(x^*,m_x)}\} \tag{3}$$

Matching Fingerprint The first step of matching fingerprint is to filter out the outliers in each trace by a threshold α. In practice, such threshold can be easily determined from a trace. For example, for the data we have shown in Figure 3, 11ms is a reasonable choice of α for CC2538, as most PRIs are within the range of $[9, 10.5]$(ms).

We then filter each trace in $\mathbb{T}$ by keeping only PRIs $\geq \alpha$:

$$T'_i = \{t \in T_i | t \geq \alpha\} \tag{4}$$

We denote the set of filtered profiling traces as:

$$\mathbb{T}'_p = \{T'_1, T'_2, ..., T'_n\} \tag{5}$$

Indeed the filtering of profiled traces could also be pre-computed.

We then apply the same filter on target trace:

$$T'_{x^*} = \{t \in T_{x^*} | t > \alpha\} \tag{6}$$

The adversary then searches for $T'_x \in \mathbb{T}'_p$ that is statistically most similar to T'_{x^*}. One way is to use Kolmogorov-Smirniov Distance (KS distance) as the measurement of statistical similarity. The adversary then outputs the index of the trace with minimum KS-distance to the target, i.e. to search T'_x such that:

$$KSD(T'_x, T'_{x^*}) = min(\{KSD(T'_i, T'_{x^*}) | i \in [1, n]\}) \tag{7}$$

where $KSD(X, Y)$ represents the KS distance between two distributions X and Y. Finally the adversary outputs x as the guess of x^*.

6.2.3 Experiment Result

We performed the experiments on CC2538. The "closed world" includes 10 applications which are:

- **powertrace**: This Contiki example continuously records the power consumption on the node.
- **broadcast**: This Contiki example broadcasts a test "Test" string periodically.
- **sensorpayload** family: These applications periodically read the sensors and report the readings to the root node encrypted. There are 8 instances of this type of applications corresponding to 8 different combinations of temperature, light and voltage sensors.

The experiments are summarised in Appendix A.

PING requests are sent from a Linux host with the command "ping6 -s 0 -i 0.4"[3]. This frequency is chosen to maximise the speed of data generation without flooding the device.

For each application, we collected 2 packet dumps (pcapng files) with each dump consists of 200000 packets. Traces of nearly 7000 PRIs are extracted from each dump.

We applied the threshold $\alpha = 11$(ms) for CC2538. Roughly 1000 PRI outliers are filtered from each trace. Table 3 summarises the data we used in the experiment.

For each trace in the experiment, there is exactly one other trace among the 20 that is collected on the same application. We applied the KS distances method as described in Section 6.2.2, as summarised in Table 4.

The experiments reported 13 out of 20 (65%) traces have successfully matched to the correct application in our setting.

6.2.4 Result Analysis

Let null hypothesis be the attack has no effect, then each trace is uniformly matched to one in the other 19 traces.

Under the null hypothesis, the probability P_0 of having 13 out of 20 traces correctly matched is hence:

$$P_0 = C_{20}^{13}(\frac{1}{19})^{13}(\frac{18}{19})^7 \approx 1.262 * 10^{-12} \leq 0.01$$

Therefore we reject the null hypothesis and conclude that the experiments have shown positive result to the attack.

7 Conclusion

In this paper, we first reported an implementation issue of 802.15.4 Security in Contiki. Then we presented examples of application of Traffic Analysis over 6LoWPAN network. The first attack shows the content of encrypted ICMP messages could be leak through the packets features such as length and MAC destination. The second attack exploits the time differences in processing a PING Request to distinguish low performance devices used in IoT applications.

We proposed a new type of side channel attack, PingProbe, in the second half of the paper. This attack exploits the outlier data in PING packets response time to fingerprint different applications running on the target node. We then conducted proof of concept experiments and proved the effectiveness of the attack under our setting.

[3]No user defined data and with interval of 0.4 second

References

[1] Online: http://http://www.contiki-os.org/.

[2] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944 (Proposed Standard), Internet Engineering Task Force, Sep. 2007, updated by RFCs 6282, 6775. [Online]. Available: http://www.ietf.org/rfc/rfc4944.txt

[3] Online: http://www.willow.co.uk/html/telosb_mote_platform.php.

[4] Online: http://www.ti.com/product/CC2538.

[5] Online: http://www.aragosystems.com/en/wisnet-item/wisnet-wismote-item.html.

[6] A. Conta, S. Deering, and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," RFC 4443 (Draft Standard), Internet Engineering Task Force, Mar. 2006, updated by RFC 4884. [Online]. Available: http://www.ietf.org/rfc/rfc4443.txt

[7] S. Chen, R. Wang, X. Wang, and K. Zhang, "Side-channel leaks in web applications: A reality today, a challenge tomorrow," in *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 2010, pp. 191–206.

[8] L. Mather and E. Oswald, "Pinpointing side-channel information leaks in web applications," *Journal of Cryptographic Engineering*, vol. 2, no. 3, pp. 161–177, 2012.

[9] S. Coull and K. Dyer, "Privacy failures in encrypted messaging services: Apple imessage and beyond," *arXiv preprint arXiv:1403.1906*, 2014.

[10] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson, "Spot me if you can: Uncovering spoken phrases in encrypted voip conversations," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 35–49.

[11] R. B. Chris Wampler, A. Selcuk Uluagac, "Information leakage in encrypted ip video traffic," ieee-globecom 2015, 2015.

[12] A. Schaub, E. Schneider, A. Hollender, V. Calasans, L. Jolie, R. Touillon, A. Heuser, S. Guilley, and O. Rioul, "Attacking suggest boxes in web applications over https using side-channel stochastic algorithms," in *Risks and Security of Internet and Systems*. Springer, 2014, pp. 116–130.

[13] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier," in *Proceedings of the 2009 ACM workshop on Cloud computing security*. ACM, 2009, pp. 31–42.

[14] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 332–346. [Online]. Available: http://dx.doi.org/10.1109/SP.2012.28

[15] C. V. Wright, S. E. Coull, and F. Monrose, "Traffic morphing: An efficient defense against statistical traffic analysis." in *NDSS*, 2009.

[16] X. Luo, P. Zhou, E. W. Chan, W. Lee, R. K. Chang, and R. Perdisci, "Httpos: Sealing information leaks with browser-side obfuscation of encrypted flows." in *NDSS*, 2011.

[17] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Protocol misidentification made easy with format-transforming encryption," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 61–72.

[18] P. Pongle and G. Chavan, "A survey: Attacks on rpl and 6lowpan in iot," in *Pervasive Computing (ICPC), 2015 International Conference on*. IEEE, 2015, pp. 1–6.

[19] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle, "6lowpan fragmentation attacks and mitigation mechanisms," in *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '13. New York, NY, USA: ACM, 2013, pp. 55–66. [Online]. Available: http://doi.acm.org/10.1145/2462096.2462107

[20] I. Krontiris, T. Giannetsos, and T. Dimitriou, "Launching a sinkhole attack in wireless sensor networks; the intruder side," in *Networking and Communications, 2008. WIMOB'08. IEEE International Conference on Wireless and Mobile Computing,*. IEEE, 2008, pp. 526–531.

[21] V. P. Singh, A. S. A. Ukey, and S. Jain, "Signal strength based hello flood attack detection and prevention in wireless sensor networks," *International Journal of Computer Applications*, vol. 62, no. 15, 2013.

[22] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Wormhole attacks in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 370–380, 2006.

[23] M. Wazid, A. Katal, R. Singh Sachan, R. Goudar, and D. P. Singh, "Detection and prevention mechanism for blackhole attack in wireless sensor network," in *Communications and Signal Processing (ICCSP), 2013 International Conference on*. IEEE, 2013, pp. 576–581.

[24] N. Sastry and D. Wagner, "Security considerations for ieee 802.15.4 networks," in *Proceedings of the 3rd ACM Workshop on Wireless Security*,

ser. WiSe '04. New York, NY, USA: ACM, 2004, pp. 32–42. [Online]. Available: http://doi.acm.org/10.1145/1023646.1023654

[25] I. . W. Group, "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," IEEE 802.15.4 Working Group, Tech. Rep., 2006. [Online]. Available: http://dx.doi.org/10.1109/ieeestd.2006.232110

[26] K.-F. Krentz, H. Rafiee, and C. Meinel, "6lowpan security: Adding compromise resilience to the 802.15.4 security sublayer," in *Proceedings of the International Workshop on Adaptive Security*, ser. ASPI '13. New York, NY, USA: ACM, 2013, pp. 1:1–1:10. [Online]. Available: http://doi.acm.org/10.1145/2523501.2523502

[27] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," RFC 4301 (Proposed Standard), Internet Engineering Task Force, Dec. 2005, updated by RFCs 6040, 7619. [Online]. Available: http://www.ietf.org/rfc/rfc4301.txt

[28] I. O. for Standardization ISO, "ISO/IEC 7498-1Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model," ISO, Tech. Rep., Jun. 1994.

[29] 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," RFC 6550 (Proposed Standard), Internet Engineering Task Force, Mar. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6550.txt

[30] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)," RFC 6775 (Proposed Standard), Internet Engineering Task Force, Nov. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6775.txt

[31] A. Conta and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," RFC 2463 (Draft Standard), Internet Engineering Task Force, Dec. 1998, obsoleted by RFC 4443. [Online]. Available: http://www.ietf.org/rfc/rfc2463.txt

[32] A. Dunkels, "The contikimac radio duty cycling protocol," *SICS Report*, 2011.

[33] Online: https://www.wireshark.org/.

A PingProbe Experiment Data Summary

Trace Index	Application	Size	Filtered Size	Note
1	broadcast	6489	593	
2	broadcast	6164	639	
3	powertrace	7142	539	
4	powertrace	7079	561	
5	Sensorpayload	7338	987	Temperature + Light
6	Sensorpayload	7963	934	Temperature + Light
7	Sensorpayload	7143	1195	Temperature only
8	Sensorpayload	7316	1096	Temperature only
9	Sensorpayload	7895	827	Light only
10	Sensorpayload	7867	789	Light only
11	Sensorpayload	7428	1138	No reading
12	Sensorpayload	7462	833	No reading
13	Sensorpayload	6565	1391	VDD only
14	Sensorpayload	7193	1111	VDD only
15	Sensorpayload	7672	955	Temperature, Light and VDD
16	Sensorpayload	7790	1023	Temperature, Light and VDD
17	Sensorpayload	7864	931	Light + VDD
18	Sensorpayload	7936	987	Light + VDD
19	Sensorpayload	7217	1222	Temperature + VDD
20	Sensorpayload	7050	1228	Temperature + VDD

Table 3: PingProbe Experiment Applications

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	N/A	**7.8**	75	78.9	48.3	58.4	19.2	15.8	62.2	58.5	34.2	34	13.2	10.3	54.1	57	52.8	58.7	19.4	13.3
2	**7.8**	N/A	81.7	85.6	54.8	65.1	25.6	22.4	68.7	65	40.8	40.4	14.9	16.7	60.7	63.7	59.3	65.4	26	19.3
3	75	81.7	N/A	**6.9**	30.3	22.9	56.3	59.5	16.2	21.2	41.2	41.5	67	65	21.5	22.8	23.2	29.8	55.8	62.5
4	78.9	85.6	**6.9**	N/A	32.3	22.5	60.3	63.6	17.9	20.9	45.2	45.6	71	69.3	25.2	22.5	26.7	27.4	59.8	66.7
5	48.3	54.8	30.3	32.3	N/A	**11.3**	29.4	32.7	15.1	14	14.5	14.7	40.1	38.3	14.8	16.1	17.1	23.8	28.9	35.6
6	58.4	65.1	22.9	22.5	11.3	N/A	39.5	42.9	9	**4.9**	24.4	24.7	50.3	48.4	14	16	16.8	23.7	39.2	45.8
7	19.2	25.6	56.3	60.3	29.4	39.5	N/A	**4.4**	43.2	39.5	19.7	17	14.5	9.6	35.3	38.5	33.7	39.7	10.8	9.6
8	15.8	22.4	59.5	63.6	32.7	42.9	**4.4**	N/A	46.6	42.8	19.5	19.1	14.9	6.2	38.4	41.7	37.1	43.2	10.9	9
9	62.2	68.7	16.2	17.9	15.1	9	43.2	46.6	N/A	**5.4**	28.2	29	54	52.1	10.3	12.4	12.3	19	42.8	49.7
10	58.5	65	21.2	20.9	14	**4.9**	39.5	42.8	5.4	N/A	24.5	25.4	50.2	48.4	11.5	13.4	14.4	20.9	39.1	46.4
11	34.2	40.8	41.2	45.2	14.5	24.4	19.7	19.5	28.2	24.5	N/A	**7.4**	28.2	24.7	20	23.1	19	24.7	15	23.9
12	34	40.4	41.5	45.6	14.7	24.7	17	19.1	29	25.4	**7.4**	N/A	25.7	23.9	20.7	23.7	19	25	14.9	21.7
13	13.2	14.9	67	71	40.1	50.3	14.5	14.9	54	50.2	28.2	25.7	N/A	11.7	45.9	49.2	44.5	50.6	15.7	**8.9**
14	10.3	16.7	65	69.3	38.3	48.4	9.6	**6.2**	52.1	48.4	24.7	23.9	11.7	N/A	44	47.2	42.8	48.7	10.5	7.2
15	54.1	60.7	21.5	25.2	14.8	14	35.3	38.4	10.3	11.5	20	20.7	45.9	44	N/A	**3.8**	6.7	11.2	34.8	41.5
16	57	63.7	22.8	22.5	16.1	16	38.5	41.7	12.4	13.4	23.1	23.7	49.2	47.2	**3.8**	N/A	8.8	8.9	37.9	44.7
17	52.8	59.3	23.2	26.7	17.1	16.8	33.7	37.1	12.3	14.4	19	19	44.5	42.8	**6.7**	8.8	N/A	11.5	33.6	40
18	58.7	65.4	29.8	27.4	23.8	23.7	39.7	43.2	19	20.9	24.7	25	50.6	48.7	11.2	**8.9**	11.5	N/A	39.4	46.1
19	19.4	26	55.8	59.8	28.9	39.2	10.8	10.9	42.8	39.1	15	14.9	15.7	10.5	34.8	37.9	33.6	39.4	N/A	**10.3**
20	13.3	19.3	62.5	66.7	35.6	45.8	9.6	9	49.7	46.4	23.9	21.7	8.9	**7.2**	41.5	44.7	40	46.1	10.3	N/A

Table 4: KS Distances of PingProbe Experiment Traces (multiplied by 100 for readability). Minimum in each row marked as **bold**.