Half Year Review

Yan Yan

April 2, 2015

1 Introduction

1.1 Background and Motivation

Cryptography is a subject studied to provide security for our data storage and transfer. Most cryptographic protocols beneath our networks today are designed based on mathematical assumptions and hence are proved to be secure [1]. However, implementations of these ”secure protocols” sometimes turn out to be insecure in real world. Techniques categorised as **Side Channel Attacks** can be used to reveal those secrets by exploiting side-channel information, such as power consumption [1] and timing [2]. Further more, recent study shows that in terms of internet, metadata in our packets, such as packet length and ACK flag, could also leak confidential information [3] [4].

The aim of this project is to further study these techniques and to analyse whether similar design flaws exist in wireless sensor networks or not, since the sensor-network has a whole different nature than the well-studied internet (low bandwidth, unreliable, etc). Further more, new types of attacks could emerge in such environment (e.g. attacks that aims at battery consumption in order to launch a denial-of-service attack).

In this project, we will be more interested information leakage through sensor-network traffic and thus is more likely to be software implementations of network protocol focused.

1.2 Activities

Other than research, I participated in the following activities:

- TA for resitting students of Cryptography A and Cryptography B 2013.
- TA for Cryptography A 2014.
- TA for System Security 2014.

[1] Technically, they are *proved* to be secure if their underlining assumptions are true.

2 Security Concerns in Sensor-networks

2.1 Implementation of Sensor-networks

The very first question of this project is to identify what kind of security protocols can be adopted in sensor-networks. Unfortunately, unlike the widely adopted internet standards (e.g. OSI model [5] and TCP/IP protocol stack), the implementing sensor-networks is still an ongoing topic due to business reason and its diverse nature of application.

So far we have done a few study on Zigbee [6] but are more interested in 6LoWPAN [7] as it is more likely to be the future standard. A noticeable feature of 6LoWPAN is that it is designed to be compatible with IPv6 which might allow us to reuse some upper layer protocols from internet in the future.

2.2 Implementation of Security Protocols on Sensor-networks

Above the diversity of sensor-network structures, implementing security protocols is even more difficult in this environment due to these constrictions:

- **Limited computational power.** Most cryptographic algorithms require heavy computational power. This contradicts the fact that most embedded systems have only very limited computational power.

- **Diverse architecture.** Real world cryptographic algorithms utilises many architecture specific optimisation due to their heavy computational requirement. Some of them also require specific instruction sets. All these facts add additional difficulty on implementations as specific code are required by different architectures.

- **Low bandwidth and unreliable transmission.** Sensor-network usually has very limited bandwidth; thus unreliable transportation protocol (i.e. protocols like UDP) is more preferable than reliable transportation protocol (i.e. protocols like TCP) as they are usually less over-headed. Most current cryptographic protocols do not handle packet lost by themselves but rely on implementation. Another problem brought by this constriction is that security protocol adds additional overhead to the packets; therefore extra care must be taken of for the trade-off between security and performance.

- **Energy intense.** Embedded system usually works on battery; hence battery life time is an important measure in such environment whilst it is rarely considered in other circumstances. Most cryptographic algorithms have tremendously impact on the power consumption of the utilising device and thus expected to be severely impactful to the life time of batteries. Key-updating also increases the power consumption. Above these, the battery itself could possibly become the new target of attacks on sensor-networks.

Despite being aware for many years, not much work we are aware of has been done on this subject. At the moment, DTLS [8] is the best candidate we have found. Some future protocols such as CoAP [9] also adopts DTLS as their security concern.

2.3 Experiment Platform

We are currently building an experimental platform in order to simulate the secure traffic on sensor-networks. We (subjectively) evaluate DTLS over 6LoWPAN to be a reasonable combination of protocol stack. We are currently using Contiki-os (`http://www.contiki-os.org/`) to provide 6LoWPAN support. However implementations for DTLS are still relatively poor at this stage that we are still trying to make a (seemingly) closest implementation called tinydtls (`http://tinydtls.sourceforge.net/`) work on our device.

Therefore our solution is to use a much less constrained device, a raspberry pi, as a proxy. The sensor is connected to the proxy through serial line and any data generated by the sensor are read and sent by the proxy. By doing this, we separate the sensor data and its transmission. The advantage is that since the transmission module is independent from the sensor node, we can easily plug-in any secure channel implementations. (We are currently using OpenSSL (`https://www.openssl.org/`).) The drawback is that this could cause the lost or fault of some important information due to the difference of hardware performance and different lower layer protocols.

3 Attack Methods against Internet

In this project, we are more interested in those attack methods that do not require physical contact to the device; i.e. everything we can obtain through a network sniffer. To be more specifically, we are more interested in packet metadata such as packet length, protocol headers, or even timestamps, instead of other physical side-channels such as power consumption.

A basic observation is that, these side-channel information are likely to be independent from the cryptographic key[2] comparing to other side-channels like power consumption [1]. Instead, these metadata are more sensitive to the data, a.k.a. message or plaintext in cryptographic term, like transferring an image file usually results in a longer packet length than a text file; therefore in this project, we are likely to be more interested in plaintext recovery rather than key recovery.

We have found some attacks on internet which exploits the side-channel information that might also be suitable in sensor-networks. It is part of our plan to analyse their practicability against sensor-networks.

- **Time-Cacheing Attack [2] [10]** This type of attack uses the variance of encryption time introduced by cache missing during s-box lookups in

[2] Except timing side-channel which could possibly lead to a key leakage. [2]

AES to recover the key. However, many architecture of sensors do not use cache so this kind of attack might not work in such environment. But the low computational power might amplify the resolution of variance; hence it might be worth to take a look at whether similar method could be developed on this scenario.

- **Mutual Information Analysis [3] [4]** Different websites have webpages with different sizes. For those with dynamic content, different user input can also result in different amount of traffic. The security protocol underneath HTTPS, i.e. TLS/SSL, does not hide this information and hence can be exploited by an adversary to recover some secret information. The efficiency of this attack is strongly affected by the entropy of input source. Sensor-networks could be vulnerable to this attack if the entropy of its data travelling through the network is low.

- **Padding Oracle Attack [11]** Some DTLS implementations verifies the padding before MAC. Since MAC computing can result in a distinct length of response time; hence it is possible to construct a padding oracle. This oracle can be used efficiently recover the plaintext when the block-cipher is working under CBC mode.

- **Compression Ratio Attack [12] [13]** Data with more repeat pattern has higher compression rate. If a malicious adversary has partial control of the message sent the victim, i.e. he can ‘’force” the victim to send some selected message alongside with some secret content, the adversary can then recover the secret content by observing the compression rate of the packets. An attack against HTTPS of such kind called BREACH [13] was shown in 2013 and is surprisingly efficient.

These are not the only attacks on internet. As a long term goal of this project, we would keep looking for such attack methods.

4 Conclusion

To conclusion, this project faces two major difficulty at this moment:

- Variation of sensor-network implementations.

- No concrete security protocol implementation.

We built an experiment platform by isolating the communication module. This gives us more flexibility but may also cause loss or inaccuracy in side-channel information.

There are several attacks developed against internet. Our next future work is to analyse their practicability on sensor-networks using our experiment platform.

References

[1] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks - revealing the secrets of smart cards.* Springer, 2007.

[2] D. J. Bernstein, "Cache-timing attacks on AES," 2004, uRL: `http://cr.yp.to/papers.html#cachetiming`.

[3] S. Chen, R. Wang, X. Wang, and K. Zhang, "Side-channel leaks in web applications: A reality today, a challenge tomorrow," in *31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berleley/Oakland, California, USA*, 2010, pp. 191–206. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/SP.2010.20

[4] L. Mather and E. Oswald, "Pinpointing side-channel information leaks in web applications," *J. Cryptographic Engineering*, vol. 2, no. 3, pp. 161–177, 2012. [Online]. Available: http://dx.doi.org/10.1007/s13389-012-0036-0

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

[6] Z. Alliance, "ZigBee specification," Tech. Rep., Jun. 2005.

[7] J. Hui and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks," RFC 6282 (Proposed Standard), Sep. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6282.txt

[8] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security," RFC 4347 (Proposed Standard), Internet Engineering Task Force, April 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4347.txt

[9] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," Internet Requests for Comment, RFC Editor, Fremont, CA, USA, Tech. Rep. 7252, Jun. 2014. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7252.txt

[10] D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: the case of aes." *IACR Cryptology ePrint Archive*, vol. 2005, p. 271, 2005. [Online]. Available: http://dblp.uni-trier.de/db/journals/iacr/iacr2005.html#OsvikST05

[11] N. J. AlFardan and K. G. Paterson, "Lucky thirteen: Breaking the tls and dtls record protocols." in *IEEE Symposium on Security and Privacy*, 2013, pp. 526–540.

[12] J. Kelsey, "Compression and information leakage of plaintext." in *FSE*, 2002, pp. 263–276.

[13] Y. Gluck, N. Harris, and A. A. Prado, "Breach: Reviving the CRIME attack," Tech. Rep., 2013. [Online]. Available: http://breachattack.com/resources/BREACH%20-%20SSL,%20gone%20in%2030%20seconds.pdf