Analyzing power consumption of TLS ciphers on an ESP32

Tilo Fischer[1], Hendrik Linka[2], Michael Rademacher[2], Karl Jonas[2], and Daniel Loebenberger[1]

[1] *Fraunhofer AISEC, Weiden i. d. OPf.*
[2] *University of Applied Sciences Bonn-Rhein-Sieg, Sankt Augustin*

April 2, 2019

More and more devices will be connected to the internet [3]. Many devices are part of the so-called Internet of Things (IoT) which contains many low-power devices often powered by a battery. These devices mainly communicate with the manufacturers back-end and deliver personal data and secrets like passwords.

With regard to security the developer of these devices are faced with the trade-off between power consumption vs. cryptography: Heavy cryptography costs computational resources which in turn decreases the time the device runs on a single battery charge. Since the latter is a feature any consumer can observe, most of the time cryptography loses.

The problem is that the power consumption of different cipher suites of a given security protocol in most real word scenarios is unknown. Miranda et al. [5] analyzed the power consumption of various Secure Sockets Layer (SSL) 3.0 and Transport Layer Security (TLS) 1.0 implementations on mobile devices. In contrast to our measurements their work focused on different implementations and not on different cipher suites of *a single* implementation. Furthermore the results are outdated because SSL 3.0 and TLS 1.0 are considered insecure nowadays. Gerez et al. [4] analyze the power consumption of TLS on a IoT device for a small subset of cryptographic functions. This paper introduced the analyses for a much lager set of cryptographic functions and additionally compared different versions of TLS with respect to power consumption. There are a few other papers [6, 1] that analyze the power consumption of TLS on low-power devices. Those only use a small subset of the supported cryptography and non of them used the new TLS version 1.3.

We employ the widespread low-cost, low-power System on a Chip (SoC) ESP32 [2] as our target device. To establish a communication channel, we used the common TLS protocol, because it is available for many platforms and analyzed by many security experts. We focused on the TLS versions 1.2 [8] and 1.3 [7]. Both versions support a plethora of cryptographic algorithms. We choose a subset of these algorithms with comparable security levels and performed a thorough power-consumption analysis.

The measurement setup consisted of a two core 240 MHz ESP32 Pico Dev Kit without voltage regulator and UART converter. WolfSSL [9] was used as the TLS client library and a laptop with an OpenSSL server served as remote

station. To measure the overall time of a 1000 Byte data transmission including the TLS protocol overhead, we used the internal time function of the micro-controller. The average deviation of this time function was found as small as 2 μs. For each cipher suite, we performed 200 independent measurements.

A Saleae Logic 16 Pro with a uCurrent was selected to measure the energy consumption. The micro-controller signaled the start and end of a measurement with a digital output. The measured current consumption and the voltage were recorded with 50 MSa/s. The resulting integral is multiplied by the median of the independent time measurements to estimate the power consumption of the respective cipher suite.

Despite the fact that these values have been obtained on a single IoT platform using a dedicated TLS implementation we obtained viable tendencies:

- Using TLS on the ESP32 requires a significant amount of energy. Compared to an unencrypted transmission, approximately 14 times more energy is required to establish a full TLS session.
- TLS 1.2 session resumption significantly reduces the required energy for IoT devices. At the moment of writing, TLS 1.3 session resumption has not been implemented.
- Using ECDSA instead of RSA for TLS signatures is beneficial in terms of energy consumption.
- TLS 1.3 further reduces the energy consumption for a full session compared to TLS 1.2.

Currently, we are aiming to extend the presented measurements with an enhanced methodology to analyze further details and include different IoT platforms and TLS implementations.

The presented knowledge in this work can be used to secure the communication of low-power devices with the lowest possible impact on the power consumption. In the best case this enables us to replace unencrypted communication with an encrypted one, which in turn improves the overall security of the IoT.

References

[1] U. Banerjee et al. "An energy-efficient reconfigurable DTLS cryptographic engine for End-to-End security in iot applications". In: (Feb. 2018), pp. 42–44. ISSN: 2376-8606.

[2] *ESP32 Series. Datasheet.* Version 2.9. Espressif Systems. Feb. 2019.

[3] D Evans. "The Internet of Things: How the Next Evolution of the Internet is Changing Everything". In: *Cisco Internet Business Solutions Group (IBSG)* 1 (Jan. 2011), pp. 1–11.

[4] A. H. Gerez et al. "Energy and Processing Demand Analysis of TLS Protocol in Internet of Things Applications". In: *2018 IEEE International Workshop on Signal Processing Systems (SiPS).* Oct. 2018, pp. 312–317. DOI: 10.1109/SiPS.2018.8598334.

[5] P. Miranda, M. Siekkinen, and H. Waris. "TLS and energy consumption on a mobile device: A measurement study". In: *2011 IEEE Symposium on Computers and Communications (ISCC)*. June 2011, pp. 983–989. DOI: 10.1109/ISCC.2011.5983970.

[6] N. R. Potlapally et al. "A study of the energy consumption characteristics of cryptographic algorithms and security protocols". In: *IEEE Transactions on Mobile Computing* 5.2 (Feb. 2006), pp. 128–143. ISSN: 1536-1233. DOI: 10.1109/TMC.2006.16.

[7] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Aug. 2018. DOI: 10.17487/RFC8446. URL: https://rfc-editor.org/rfc/rfc8446.txt.

[8] Eric Rescorla and Tim Dierks. *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246. Aug. 2008. DOI: 10.17487/RFC5246. URL: https://rfc-editor.org/rfc/rfc5246.txt.

[9] *wolfSSL Embedded SSL/TLS Library — Now Supporting TLS 1.3*. URL: https://www.wolfssl.com/ (visited on 03/26/2019).