Adaptive Optimization of TLS Overhead for Wireless Communication in Critical Infrastructure

Jörn Bodenhausen*, Laurenz Grote*, Michael Rademacher‡§, Martin Henze*§

*Security and Privacy in Industrial Cooperation, RWTH Aachen University, Germany
‡University of Applied Sciences Bonn-Rhein-Sieg, Germany
§Cyber Analysis & Defense, Fraunhofer FKIE, Germany
{bodenhausen, henze}@spice.rwth-aachen.de · laurenz.grote@rwth-aachen.de · michael.rademacher@h-brs.de

Abstract—**With critical infrastructure increasingly relying on wireless communication, using end-to-end security such as TLS becomes imperative. However, TLS introduces significant overhead for resource-constrained devices and networks prevalent in critical infrastructure. In this paper, we propose to leverage the degrees of freedom in configuring TLS to dynamically adapt algorithms, parameters, and other settings to best meet the currently occurring resource and security constraints in a wireless communication scenario. Consequently, we can make the best use of scarce resources to provide tightened security for wireless networks in critical infrastructure.**

Index Terms—**Transport Layer Security, Wireless Networks, Critical Infrastructure.**

I. Motivation

Modern critical infrastructure is increasingly interconnected, while simultaneously being deployed over significantly larger areas [1]. Examples of this trend range from wind parks over power grids to smart cities. However, the special characteristics of such widespread environments often render wired communication infeasible and thus call for the use of cost-efficient wireless network technology such as the 450 MHz LTE-M network for critical infrastructure in Germany [2].

However, the shift to wireless communication as visualized in Figure 1 comes with severe security implications and challenges. Even though the assumption of security through physical separation of a network ① has long been invalid due to their connection to the Internet ②, deployments are still not adequately protected [3], [4]. For wireless technology, the implications are even more profound, as security goals are particularly easy to compromise even for private infrastructure ③ [5], [6]. Furthermore, wireless network infrastructure is commonly provided by a cellular operator and thus shared with other entities in a dedicated network ④ or even entirely public and routed via the Internet ⑤. Thus, any communication potentially traverses third-party infrastructures.

The most promising approach to address resulting security concerns is end-to-end security, even if other security mechanisms are in place [7]. In fact, regulators often demand the use of TLS, the most prominent end-to-end security approach, for communication in critical infrastructure, especially when using wireless communication [2], [8]. However, besides all advantages such as flexibility and interoperability, the use of TLS can constitute significant overhead for resource-constrained devices and networks [8]. Still, and providing the
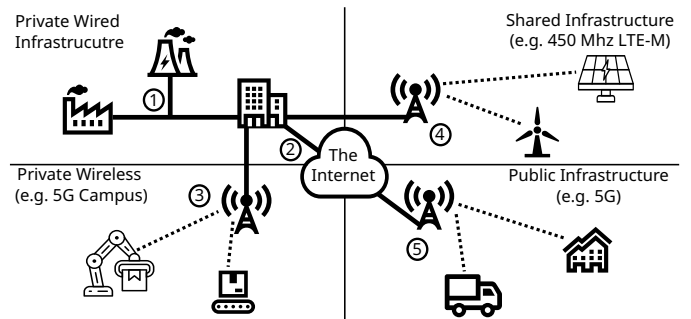

Fig. 1. With critical infrastructure becoming more widespread and interconnected, a shift from traditional wired networks ①, ② to wireless networks with private ③, shared ④, and public ⑤ infrastructure becomes necessary.

main motivation for this work, this overhead is not static as it depends on concrete parameterization, opening the potential to optimize the TLS overhead for specific scenarios.

Related Work. Various works study the overhead of TLS through measurements, e.g., with regard to (i) constrained LoRaWAN networks [8], (ii) energy [9], or (iii) CPU, memory, and bandwidth overhead in TLS-secured MQTT [10]. While these works only focus on a particular setting or limited set of parameters, they still indicate that a trade-off for particular resources is possible. For example, Restuccia et al. [11] identify memory overhead variations between TLS and DTLS as well as different implementations. Moreover, for post-quantum algorithms, depending on the used network technology, either bandwidth or computational time is the main limiting factor for TLS connection establishment [12]. Furthermore, proposed optimization efforts are generally focused on particular settings. For instance, Lauer et al. [13] focus on optimizing cryptographic computations on hardware-accelerated devices.

Contributions. To adaptively optimize TLS for wireless communication in critical infrastructure and thus enable its widespread use even in challenging scenarios, we propose a two-step approach. First, to obtain a thorough understanding of the TLS overhead, we perform comprehensive measurements along several dimensions, covering all potentially practically relevant settings and a multitude of algorithms and parameters (Sec. II). Second, we turn these insights into use by designing and implementing an approach that can dynamically choose and adjust TLS parameters to meet resource constraints of
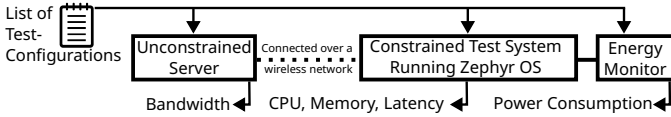
Fig. 2. Our comprehensive measurement setup enables the analysis of the overhead of TLS across various dimensions.

a given scenario and adapt to changes, e.g., in available bandwidth (Sec. III). Moreover, to illustrate how this approach can be utilized and implemented in real-world deployments, we describe its application to a particular use-case scenario and demonstrate the significance of TLS bandwidth optimization in this scenario through practical measurements (Sec. IV).

II. Variability in the Overhead of TLS

Any efforts towards optimization of TLS require a profound understanding of the resulting overhead and its variability under certain conditions. While related work already provides valuable insights into isolated aspects of TLS overhead, a comprehensive picture of all influence factors across all relevant dimensions of TLS overhead is still missing. To fill this gap, we propose a comprehensive measurement setup to study variability in TLS overhead (Sec. II-A) and then exemplary showcase the potential for optimizing the bandwidth consumption of TLS especially in wireless networks (Sec. II-B).

A. Comprehensive Measurement Setup

To lay the foundation to thoroughly study the impact of various TLS configurations, we outline the design of a comprehensive measurement setup to evaluate the TLS overhead across various dimensions. Our intended setup is illustrated in Figure 2 and consists of an unconstrained server, a constrained system that shall be evaluated, as well as an energy monitor. Initially, the devices are provided with a list of TLS configurations that shall be tested. The constrained device then iterates over the list and connects to the server with the respective configuration while the overhead is measured on the respective devices. Bandwidth is evaluated on the unconstrained server, as the client might not be capable of creating network captures. CPU, memory, and latency overhead must be evaluated directly on the constrained device. For the evaluation of power consumption, additional hardware is required similar to the setup from Suárez-Albela et al. [9]. While we intend to evaluate each overhead dimension independently to avoid interference, our proposed setup promises to provide valuable insights into their interdependency and to identify trade-off potential.

B. Case Study: Bandwidth Overhead

To better illustrate the potential of the proposed comprehensive measurement setup, we will exemplary showcase the huge variability in TLS bandwidth overhead depending on the TLS configuration. More specifically, we study the bandwidth required for different TLS versions and authentication mechanisms as well as varying elliptic curves. To this end, we utilize a custom measurement setup that can capture local TLS communication generated by OpenSSL 3.2.1, wolfSSL
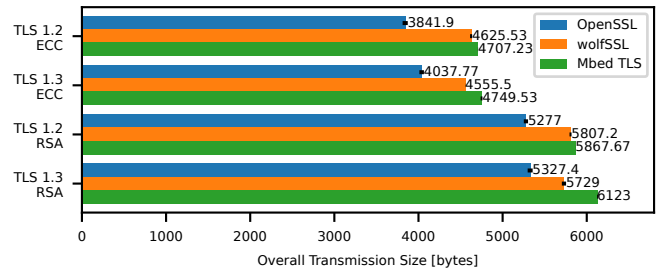

Fig. 3. The average bandwidth overhead of a full TLS handshake varies widely across different authentication mechanisms as well as TLS libraries.
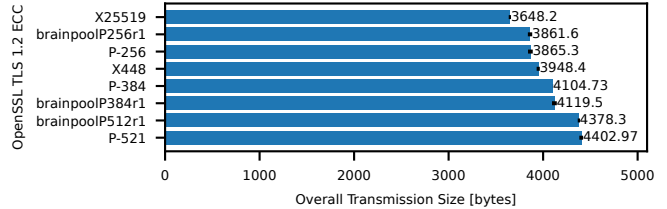

Fig. 4. While the bandwidth overhead of TLS messages mainly depends on the security level of the used ECC curve, small variations exist even for curves with the same security.

5.6.6, or Mbed TLS 3.6.0 respectively. For each parameter combination, we perform 30 runs and the observed variance can be traced back to varying acknowledgements and negligible variance in the generated keys and signatures. We perform a full handshake with mutual authentication via a single self-signed X.509 certificate as generated by OpenSSL and subsequently exchange two 128 byte messages. In the following, we report on the arithmetic mean over the runs of each setting and their standard deviation.

A first measurement illustrated in Figure 3 highlights the significance of the authentication mechanism on the overall bandwidth overhead. Moreover, a significant difference between libraries due to different default configurations and message processing can be observed. Our second measurement illustrated in Figure 4 considers various elliptic curves used within an *ECDHE_ECDSA* connection establishment. Since various TLS messages are influenced by this particular choice, elliptic curves are an excellent example of a particular configuration choice that yields a trade-off potential between bandwidth overhead, security level, and further dimensions not yet examined by us (e.g., energy [9]).

Our preliminary results provide several insights. First, there is indeed a huge potential to optimize TLS overhead, especially for bandwidth-constrained communication. This can already be seen from the varying bandwidth between default configurations of libraries for a fixed key-exchange method. Moreover, a trade-off between security and bandwidth is possible, as an increase in security due to larger key sizes directly results in additional bandwidth overhead. Lastly, comprehensive measurements are required to uncover variances and differences between different implementations and scenarios.
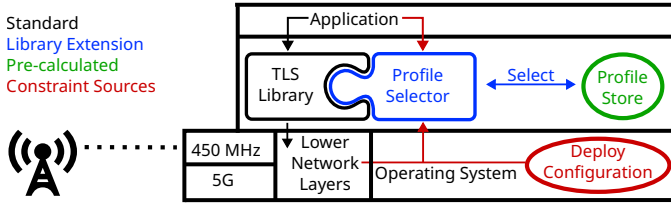
Fig. 5. We propose to extend a TLS library with a profile selector that chooses algorithms and parameters tailored to current resource constraints based on pre-computed profiles.
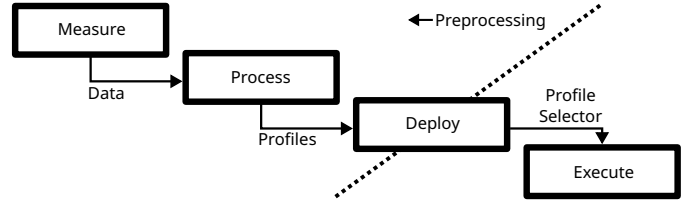

Fig. 6. Much of the additional overhead, in particular the generation of profiles, would be performed in a pre-processing phase before the actual deployment, resulting in manageable processing overhead at run-time.

III. ADAPTIVE TLS OVERHEAD OPTIMIZATION

The overhead of TLS is not static, but depends on the particular configuration and parameters, allowing for a trade-off between different optimization dimensions [11], [12]. To leverage this potential, we propose to automatically adapt TLS parameters to achieve the best trade-off for the current resource and security constraints.

Our basic concept for such adaptive TLS optimizations is illustrated in Figure 5. Without our proposed optimizations, an application running on a constrained platform has to use a TLS library with a configuration that is chosen and fixed at compile time (black), preventing dynamic adaptations to resource constraints. Our main idea to extend this setting with the ability to dynamically adapt the TLS configuration to changing resource-constraints is to extend the TLS library with a *profile selector* (blue). Without introducing breaking changes to the interaction between applications and libraries (thus affording backwards compatibility), the profile selector applies the best-fitting *profile* (Sec. III-A), i.e., pre-defined configuration of a TLS library, based on the currently given *resource and security constraints* (Sec. III-B).

A. Profiles of TLS Configurations

Profiles provide a collection of algorithms, parameters, and settings that represent a configuration of the TLS library together with information on the resulting overhead across all relevant dimensions and the security level. In particular, profiles will be generated for and adjusted to a given device and network in the development phase and deployed onto the device. During run-time, whenever changes in constraints are observed, the profile selector will select the best-fitting profile based on the new set of constraints. The resulting precise tailoring to a particular setting promises to enable near-optimal utilization of the trade-off and optimization potential of TLS configurations. Moreover, due to the pre-calculation of profiles, the run-time overhead of our approach is minimized.

B. Dynamic Resource and Security Constraints

Various sources of constraints have to be considered, most of which are dynamic, e.g., as they depend on current system utilization or condition of the wireless link. These dynamics are immense as they are directly influenced by the current state of the underlying network, which the application is usually not aware of. For example, in wireless networks, a connection may use a high-order modulation when a certain signal strength is

available [14]. In other situations, a bad reception can significantly decrease throughput and increase the packet error rate, rendering timely transmission of TLS handshakes infeasible [8]. Furthermore, traffic patterns in critical infrastructure can be highly dynamic due to random events such as natural disasters, updates, or cyberattacks [15]. In such events, a huge amount of devices may initiate simultaneous transmission, causing a shortage of radio resources or even interference.

To account for such challenges, the current TLS profile needs to be adapted dynamically based on these changing constraints. Depending on the type of constraint, such updates can either be provided by the application or the operating system (cf. Figure 5). For example, the application can provide meta-data regarding the urgency of the transmission and its security requirements. Likewise, the operating system can extract information about the current reception from the modem. While, in general, a single device has no information about the current or future overall network load, it can partially detect an overload situation due to an increased packet error rate or different transport layer information (e.g., TCP) and may defer its transmission or switch to another TLS profile.

IV. USE-CASE EXAMPLE

While our measurement setup (cf. Sec. II) promises to bring valuable and comprehensive insights into the interplay between TLS overhead in various dimensions and our profile selection mechanism (cf. Sec. III) aims to benefit from such insights, so far it remains open how this can be utilized in practice. Hence, in this section, we describe a use-case example to illustrate how those components can be integrated to result in real-world savings. For this, we consider a critical infrastructure network that monitors and controls distributed energy systems wirelessly, e.g., by periodically reporting sensor measurements [16], [17]. We assume it to mainly utilize the 450 Mhz LTE-M network [2], with 5G serving as a backup. This results in potential switching between shared and public infrastructure (cf. Figure 1), which could lead to varying security requirements and resource constraints due to technical limitations of these technologies. Moreover, available bandwidth for each technology might fluctuate depending on the situation (cf. Sec. III-B).

In such a setting, detailed measurements can yield interesting insights to improve efficiency at run-time. For instance, detailed measurements of the reception of the 450 Mhz network throughout a building allow for insights into attenuation effects
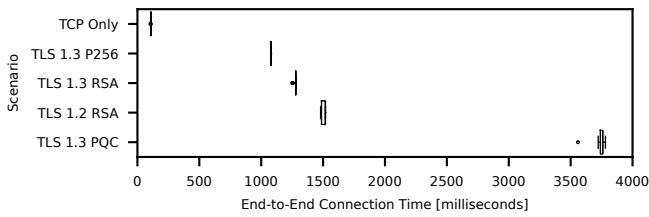
Fig. 7. Introducing additional overhead due to large TLS handshakes into an LTE-M network leads to significant latency degradation.

[18]. At run-time, such information can be utilized to estimate the available bandwidth at a certain deployment location. The different steps required to make such information usable within our proposed profile selection mechanism is illustrated in Figure 6. The foundation is formed by detailed measurements in the lab and in the wild for various TLS configurations. In a subsequent analysis phase, the results are examined in detail and insights about potential trade-offs are derived. From this, a suitable set of profiles is generated once and integrated in the profile selector at compile time (though later updates are also possible). Thus, at run-time, merely a selection of suitable profiles needs to be performed by searching the profile store. The data depicted in Figures 3 and 4 already shows that a trade-off between security and bandwidth is possible, i.e., striving for the most secure key exchange mechanisms as long as the available bandwidth allows for it.

To further illustrate the significance of the bandwidth over-head in particular on the overall performance of such a network, we performed a series of measurements. The results of these measurements are shown in the boxplot in Figure 7. Here, we performed load tests on the 450 Mhz LTE-M Network between an Amarisoft Callbox and an Amarisoft UE Simbox simulating 64 devices. We emulate polling of electrical substations with the IEC 60870-5-104 protocol, i.e., a 38 bytes request followed by a 282 bytes reply, in one second intervals. Thus, the TLS handshake constitutes a significant overhead. We observe that the achievable end-to-end connection time in the network significantly degrades with the introduction of TLS and further degrades when utilizing post-quantum secure mechanisms. Here, e.g., an opportunistic use of post-quantum mechanisms would be conceivable with a fallback to conventional mechanisms in case of an overload situation to ensure availability.

While this illustrates that a trade-off generally is possible and can be utilized in a real-world scenario, focussing on security and bandwidth alone might not yield groundbreaking efficiency optimization on its own. However, the possibility to benefit from further trade-off potential is extremely likely, as for instance highlighted by related work for computational overhead of certain post-quantum secure algorithms [12] or energy consumption [9], which is particularly relevant for battery-powered devices. Furthermore, such considerations for optimization can be extended to the timing of handshakes and the utilization of session resumption mechanisms.

V. OUTLOOK & CONCLUSION

With our work, we strive to leverage the degrees of freedom in configuring TLS to dynamically adapt its configuration to best meet varying resource and security constraints, especially for wireless communication in critical infrastructure. Currently, we are working on fully realizing the comprehensive measurements of variability in TLS overhead (Sec. II) and implementing our concept of adaptive TLS overhead optimization (Sec. III). To this end, we are joining forces with a device manufacturer to integrate our solution into a commercial sensor platform for critical infrastructure that unites 450 MHz LTE-M and 5G communication capabilities, which will allow us to evaluate our approach in a real-world 450 MHz LTE-M network of a cooperating utility provider. Already today, our case study results show huge potential for adaptive optimization of TLS overhead for wireless communication in critical infrastructure (Sec. IV). With our concept for adaptive TLS overhead optimization, we outline an approach to leverage this potential with minimal changes to existing devices and applications, thus affording wide deployability.

REFERENCES

[1] M. El-Hajj et al., "Analysis of authentication techniques in Internet of Things (IoT)," in CSNet, 2017.
[2] J. Bodenhausen et al., "Securing Wireless Communication in Critical Infrastructure: Challenges and Opportunities," in MobiQuitous, 2023.
[3] M. Dahlmanns et al., "Missed Opportunities: Measuring the Untapped TLS Support in the Industrial Internet of Things," in ASIACCS, 2022.
[4] T. Krause et al., "Cybersecurity in Power Grids: Challenges and Opportunities," Sensors, vol. 21, no. 18, 2021.
[5] O. A. Osanaiye et al., "Denial of Service Defence for Resource Availability in Wireless Sensor Networks," IEEE Access, vol. 6, 2018.
[6] S. Michaelides et al., "Secure integration of 5g in industrial networks: State of the art, challenges and opportunities," arXiv:2408.16833, 2024.
[7] M. Henze et al., "A Trust Point-based Security Architecture for Sensor Data in the Cloud," in Trusted Cloud Computing, 2014.
[8] M. Rademacher et al., "Bounds for the Scalability of TLS over Lo-RaWAN," in ITG-Symposium Mobile Communication, 2022.
[9] M. Suárez-Albela et al., "A Practical Performance Comparison of ECC and RSA for Resource-Constrained IoT Devices," in GIoTS, 2018.
[10] V. Dimov et al., "Resource tradeoffs for TLS-secured MQTT-based IoT Management," in NOMS, 2022.
[11] G. Restuccia et al., "Low-Power IoT Communication Security: On the Performance of DTLS and TLS 1.3," in PEMWN, 2020.
[12] R. Gonzalez and T. Wiggers, "KEMTLS vs. Post-quantum TLS: Performance on Embedded Systems," in SPACE, 2022.
[13] F. Lauer et al., "Analysis and Optimization of TLS-based Security Mechanisms for Low Power IoT Systems," in CCGRID, 2020.
[14] M. Rademacher et al., "Path Loss in Urban LoRa Networks: A Large-Scale Measurement Study," in VTC2021-Fall, 2021.
[15] Ö. Sen et al., "On Using Contextual Correlation to Detect Multi-stage Cyber Attacks in Smart Grids," SEGAN, vol. 32, 2022.
[16] M. Eggert et al., "SensorCloud: Towards the Interdisciplinary Development of a Trustworthy Platform for Globally Interconnected Sensors and Actuators," in Trusted Cloud Computing, 2014.
[17] M. Henze et al., "Maintaining User Control While Storing and Processing Sensor Data in the Cloud," IJGHPC, vol. 5, no. 4, 2013.
[18] C. Sorgatz et al., "Measurements of Building Attenuation in 450 MHz LTE Networks," in Mobilkommunikation; 28. ITG-Fachtagung, 2024.