Poster: Transport Security Orchestration Using DNS

David Heye*, Sahi Islam*,†, Jan Pennekamp*, Klaus Wehrle*
*Communication and Distributed Systems, RWTH Aachen University, Germany · {lastname}@comsys.rwth-aachen.de
†University of Alberta, Canada · sahi2@ualberta.ca

***Abstract*—Communication networks enable the exchange of data with varying sensitivity, from non-sensitive public files to highly confidential healthcare or financial records. Cryptographic protection introduces significant computational and communication overhead. While lightweight ciphers have been proposed to reduce this burden, they compromise security and are unsuitable for sensitive data. We propose a system that enables adaptive security by embedding service sensitivity information in the Domain Name System (DNS), allowing peers to select appropriate cryptographic primitives based on data requirements. This approach ensures adequate protection while minimizing overhead. Additionally, it can be seamlessly integrated into existing networks without additional hardware. Initial results indicate improved throughput and reduced computational load on hosts.**

I. Introduction

Modern networks, including the Internet, are used to transmit a variety of data, ranging from non-sensitive telemetry values, e.g., from smart-homes, to public social-media traffic to highly-sensitive financial or healthcare records. Each of these types of data imposes distinct requirements for confidentiality (privacy) and integrity. Commonly, devices use a one-size-fits-all approach where data is protected using the highest level of security at all times [1]–[3]. On the one hand, this approach safeguards the data while introducing unnecessary, computation-induced overhead since stronger encryptions usually implies more complex computations. Consequently, this approach oftentimes increases communication latency and energy consumption for non-sensitive traffic. On the other hand, manually configuring per-host or per-service policies is error-prone and does not scale in multi-domain environments.

We, therefore, propose a lightweight and performant Domain Name System (DNS)-inspired orchestration mechanism in which each host publishes its required security level for different services. Before establishing connections, clients query the recipients' security profile and then initiate sessions using those parameters. In particular, this design enables a centralized management of security policies, maintaining compatibility with legacy devices, while eliminating the need for complex and more error-prone decentralized policy updates.

II. Motivation and Background

The proliferation of interconnected systems, ranging from cloud platforms and Internet of Things (IoT) devices to distributed industrial control networks, has fundamentally changed how data is generated, transmitted, and consumed. These systems increasingly exchange sensitive information across diverse and often untrusted networks. Hence, ensuring the confidentiality, integrity, and authenticity of data through cryptographic mechanisms is now standard practice [4], most prominently using Transport Layer Security (TLS) [3] and Datagram Transport Layer Security (DTLS) [2].

However, these security measures introduce non-negligible overheads: On constrained devices, cryptographic computations can significantly impact system responsiveness and energy consumption [5]. Moreover, even powerful machines may face reduced throughput and increased operating costs [6]. Although the performance can be improved using, e.g., hardware acceleration, the aggregate overhead remains substantial at scale. Related work suggests deploying lightweight ciphers to reduce computational overhead [5], [7] or Message Authentication Code (MAC) aggregation to reduce communication overhead [8]. These approaches, however, typically involve tradeoffs in security strength, making them unsuitable for highly sensitive data where strong guarantees are essential. These circumstances raise the question of alternative means for balancing security requirements with system capabilities.

Despite the diversity of applications, current practice largely applies a uniform level of security, possibly due to the widespread and standardized adoption of standardized cipher suites in protocols like TLS and DTLS [2], [3]. For example, the download of a Linux distribution is typically non-sensitive but safeguarded with the same cryptographic strength as the access to highly sensitive information, e.g., online banking or healthcare records. This situation extends to other domains, including public sensor data in smart cities, where confidentiality is not a concern. Nonetheless, ensuring integrity of such information remains important to prevent any tampering.

These examples motivate the need for adaptive security approaches that adjust security levels based on the sensitivity and context of information. Such strategies may avoid unnecessary computational overheads and energy consumption by preventing overprovisioning of security where it is not needed.

III. Design: DNS-Based Security Orchestration

To address this gap, we propose storing data security policies that enable entities to apply appropriate security building blocks when accessing the data. Specifically, we leverage the well-established DNS [9], [10] for the distribution of the policies. This approach allows our solution to be seamlessly integrated into existing infrastructures without requiring additional hardware or dedicated services.

We define a new DNS record type `SEC` for capturing the security requirements of a given service. The structure of this type is inspired by the `SRV` record, which provides details such

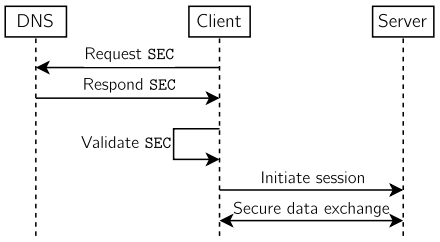

Figure 1. Simplified view of a client establishing a communication session with a server. The client first requests the required security level from the DNS service, e.g., whether to employ TLS at all, from the SEC record. Afterward, client and server establish a session with the server using cryptographic primitives that match the requirements learned from the SEC record.

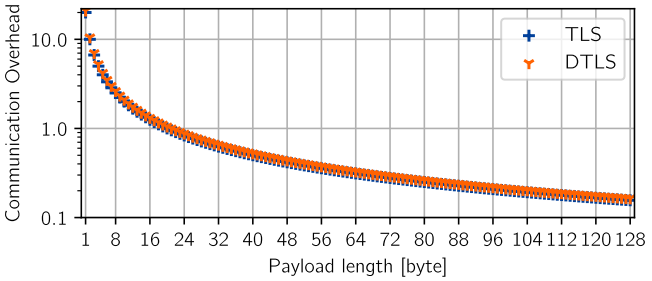

Figure 2. Communication overhead induced by TLS and DTLS for short messages with payloads between 1 B and 128 B. The overhead accounts for headers and authentication tags that are added by the respective protocols.

as port numbers used by a service. As the generic TXT record is prone to misuse and lacks standardized semantics [11], the dedicated SEC type ensures clarity and prevents ambiguity.

As the security level conveyed by a SEC record is intentionally public, it does not require encryption. However, for safeguarding purposes, requesters should be able to validate both the authenticity and integrity of these records. Our design achieves this need by mandating and including a strong digital signature, generated by the service provider, for each record.

Figure 1 shows a simplified overview of the connection establishment. The client verifies the authenticity of the signed DNS response upon reception. If the information is authentic, the client follows the specified level of security during connection establishment. If the security level mandates the use of cryptography, e.g., it establishes a TLS or DTLS session solely advertising adequate cipher suites. For non-sensitive transmissions, the peers can fall back to insecure but undemanding transport protocols like TCP [12] or UDP [13], avoiding unnecessary cryptography-induced overheads.

IV. First Evaluation Results

To assess the practicality of our approach in real-world scenarios, we conduct initial measurements in a Mininet [14] network simulation. We model a company network with multiple hosts connected via switched and routed networks, enabling diverse communication patterns. Here, we define multiple data classes with different sensitivity. Before connecting to a server, query a central entity, the DNS, to obtain information about the sensitivity of the requested service.

In our current system, we distinguish between three types of sensitivity: (i) data that does not require any security (using only TCP for transport), (ii) data that only requires integrity protection and authenticity (using TLS with an integrity-only cipher suite [15]), and (iii) data that requires encryption, integrity protection, and authenticity (using traditional TLS).

Table I
Throughput of wolfSSL [6], [16] for different ciphers on different hardware. As expected, only providing integrity (HMAC-SHA) yields higher throughput than combining integrity protection with confidentiality (AES-GCM). Any type of security induces a non-negligible overhead.

Hardware	Algorithm	Throughput
AMD EPYC 7443	AES-128-GCM	19.68 MB/s
	AES-256-GCM	17.27 MB/s
	HMAC-SHA-256	40.97 MB/s
ESP32-WROOM-32	AES-128-GCM	331.75 kB/s
	AES-256-GCM	312.20 kB/s
	HMAC-SHA-256	1733.0 kB/s

Our measurements indicate that in case (i), the communication overhead induced by queries to the DNS is negligible due to the absence of cryptographic protocol overhead. Figure 2 emphasizes the amount of communication overhead introduced by TLS and DTLS for short messages. In case (ii), the transmitted data volume remains comparable to traditional TLS. However, the individual hosts benefit from reduced computational load as they no not perform any encryption or decryption as indicated by the throughput measurements from Table I. Case (iii) serves as our baseline as this is how communication security is usually handled in current networks. Here, the additional queries to the DNS introduce some additional communication before establishing a session, however, this overhead can be minimized by leveraging caching techniques.

V. Conclusion and Future Work

Overall, our proposed design of an adaptive transport security orchestration reveals promising results. First, it demonstrates reduced workload on network hosts and decreases the network overhead for non-sensitive data without underprovisioning security. We are currently looking into the exact implementation of the SEC record to ensure secure dissemination of security levels from service providers to clients. Specifically, we plan to incorporate short but robust digital signatures and are developing an efficient validation scheme.

Next, we will survey the implications of our design on resource-constrained hardware to give a better account of their experienced benefits. We further would like to look into a mechanism that enables adjusting the security level between peers on the fly, without interrupting an established communication session. Given some limitations encountered with Mininet, we intend to complement our evaluation with either event-based network simulations or real-world performance measurements to provide a more comprehensive assessment.

We are confident that these evaluations will further underline the advantages of our DNS-based design and represent an important step toward more sustainable yet secure networking.

Acknowledgments: This work was funded by the Federal Ministry of Research, Technology and Space (BMFTR) in Germany under the grant number 16KIS2251 of the SUSTAINET-guardian project. The responsibility for the content of this publication lies with the authors.

REFERENCES

[1] R. T. Fielding, M. Nottingham, and J. Reschke, "HTTP Semantics," RFC 9110, Jun. 2022.

[2] E. Rescorla, H. Tschofenig, and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3," RFC 9147, Apr. 2022.

[3] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018.

[4] K. Scarfone, M. Scholl, and M. Souppaya, "Security Considerations for Exchanging Files Over the Internet," Aug. 2020.

[5] M. Rana, Q. Mamun, and R. Islam, "Balancing Security and Efficiency: A Power Consumption Analysis of a Lightweight Block Cipher," *Electronics*, vol. 13, no. 21, Nov. 2024.

[6] wolfSSL, "Benchmarking wolfSSL and wolfCrypt," Accessed: 2025-08-14. [Online]. Available: https://www.wolfssl.com/docs/benchmarks/

[7] I. Bhardwaj, A. Kumar, and M. Bansal, "A review on lightweight cryptography algorithms for data security and authentication in IoTs," in *2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*, Sep. 2017, pp. 504–509.

[8] E. Wagner, D. Heye, J. Bauer, K. Wehrle, and M. Serror, "MAC Aggregation over Lossy Channels in DTLS 1.3," in *Proceedings of the 33rd IEEE International Conference on Network Protocols*, 2025.

[9] "Domain names - concepts and facilities," RFC 1034, Nov. 1987.

[10] "Domain names - implementation and specification," RFC 1035, Nov. 1987.

[11] A. Portier, H. Carter, and C. Lever, "Security in Plain TXT — Observing the Use of DNS TXT Records in the Wild," in *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, Jun. 2019, pp. 374–395.

[12] W. Eddy, "Transmission Control Protocol (TCP)," RFC 9293, Aug. 2022.

[13] "User Datagram Protocol," RFC 768, Aug. 1980.

[14] Mininet, "Mininet: Rapid Prototyping for Software Defined Networks," Accessed: 2025-08-14. [Online]. Available: https://mininet.org/

[15] N. Cam-Winget and J. Visoky, "TLS 1.3 Authentication and Integrity-Only Cipher Suites," RFC 9150, Apr. 2022.

[16] wolfSSL, "wolfSSL Embedded SSL/TLS Library," Accessed: 2025-08-14. [Online]. Available: https://wolfssl.com/products/wolfssl