# A Certificate-centric Security Oracle for the Trustworthy Internet

Hyunsoo Kim
wayles@snu.ac.kr
Seoul National University
Seoul, Republic of Korea

Sangwi Kang
ksw292002@snu.ac.kr
Seoul National University
Seoul, Republic of Korea

Kyunghan Lee
kyunghanlee@snu.ac.kr
Seoul National University
Seoul, Republic of Korea

Taekyoung "Ted" Kwon
tkkwon@snu.ac.kr
Seoul National University
Seoul, Republic of Korea

## Abstract

The Internet's current security landscape remains fundamentally compromised by inherent protocol vulnerabilities. Existing security solutions have been narrowly focused, addressing isolated protocols or specific attack vectors without developing a comprehensive security framework. Our proposed certificate-centric security oracle represents a paradigm shift, offering universal security services that enable protocols to robustly attest their identities and authentically validate their communications. The security oracle operates in a cross-layer fashion so that any protocol (in any layer) that wishes to use such services can be extended individually with minor efforts (say, adding a few fields to carry a certificate or a signature). We also present a few case studies illustrating how existing protocols can be extended to use the services of the security oracle. Prototype-based experiments are carried out for ICMP and DHCP to demonstrate the practical feasibility of the proposed framework.

## CCS Concepts

• **Security and privacy** → **Security protocols**; *Web protocol security*; Digital signatures; Authentication.

## Keywords

Security Oracle, Internet Protocols, Authentication, Certificate

**ACM Reference Format:**
Hyunsoo Kim, Sangwi Kang, Kyunghan Lee, and Taekyoung "Ted" Kwon. 2025. A Certificate-centric Security Oracle for the Trustworthy Internet. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3701716.3715493

## 1 Introduction

As the Internet originated as an experimental network, security was not considered in its initial design. Over the past few decades, it has evolved into a critical infrastructure. However, its security and trustworthiness still fall short of expectations. Various attacks, such as DNS cache poisoning, man-in-the-middle intrusions, BGP IP prefix hijacking, and rogue WiFi access points, demonstrate the vulnerabilities that persist in current Internet operations.

In response to these threats, numerous point solutions have been proposed and some of them are deployed to enhance the security of a specific protocol or to mitigate a particular attack. For example, IPsec [10] enables encrypted IP-layer communication but requires manual provisioning, restricting its deployment to specific environments. DNSSEC [4] introduces a hierarchical key chain for DNS records, but its adoption has been slow due to challenges like the distribution of `Delegation Signer` records [3]. TLS [9] allows clients to authenticate servers and supports end-to-end encryption, but it is limited to application layer processes. Overall, no holistic security framework exists that can be applied across multiple layers and protocols.

Clean-slate architectures, such as XIA [2] and SCION [8], have not achieved widespread deployment because they significantly deviate from the current TCP/IP infrastructure and lack convincing transition paths for network operators [1].

To address these issues, we propose a unified security framework that enables individual Internet protocols to incorporate security services as needed. The core of this framework is—a *Security Oracle*—that interacts with protocols across the data, control, and management planes. By invoking the Security Oracle, any protocol instance can request authentication and other security-related services, as described in the following section.

Certificates serve as the fundamental elements of the Security Oracle. Each network entity—whether a host, server, switch, router, or other device—can hold multiple certificates, each attesting to a particular identity or an attribute (e.g., an IP address for ICMP or a domain name for a web server). When a protocol like ICMP leverages the Security Oracle, it can append a certificate verifying its IP address and include a corresponding digital signature with its message, thereby enabling recipients to authenticate the sender's claimed identity and confirm the message's integrity during transmission.

Once its certificate and signature are verified, the protocol instances can optionally perform a key exchange to establish a shared session key for encryption and decryption. While some protocols may only require authentication and integrity, the capability of making an encrypted session allows protocols to offer confidentiality when desired, depending on user preferences or network configurations.

The contributions of this paper are summarized as follows.



*WWW Companion '25, April 28-May 2, 2025, Sydney, NSW, Australia*

ACM ISBN 979-8-4007-1331-6/25/04
https://doi.org/10.1145/3701716.3715493

- We introduce a Security Oracle that provides security services such as certificate management/validation, signature generation/verification, and encryption/decryption.
- We show that any existing protocol can be extended to incorporate certificate exchange and signatures using the Security Oracle's services.
- Although the proposed framework requires extending current protocols, it does not mandate a clean-slate redesign. We present case studies illustrating how existing protocols can benefit from this approach.

## 2 Certificate-centric Security Oracle

### 2.1 Overview

A clean-slate approach to security is challenging due to the complexity of replacing legacy protocols or maintaining backward compatibility. While the IETF has standardized several security solutions (e.g., DNSSEC, RPKI[6], IPsec), their deployment remains slow or provides a solution only for a specific protocol. In contrast, we propose a holistic framework, *Security Oracle*, which provides a suite of security services to any protocol with its own extension.

As illustrated in Figure 1, we introduce a Security Oracle into the Internet protocol stack, which traditionally includes data, control, and management planes (We focus on the data plane for simplicity). Any data-plane protocol can request security services from the oracle, which is accessible at various layers (kernel, device drivers, or applications).

From the analysis of legacy security protocols, including TLS, IPsec, RPKI, and DNSSEC, we identify the following security functions:

- *Key/Certificate Storage*: Stores the entity's own certificate(s) and private key(s), as well as trusted root CA certificates for chain validation.
- *Certificate Validation*: Validates incoming certificate chains and checks certificate revocation status, typically by querying issuing CAs.
- *Signature Verification*: Verifies digital signatures using the counterpart's public key extracted from its certificate.
- *Signature Generation*: Produces digital signatures using the entity's private key, associated with a corresponding certificate.
- *Shared Key Generation*: Derives a shared key (e.g., via Diffie-Hellman exchanges) to establish a secure session in which the following packets are encrypted and decrypted.
- *Encryption/Decryption*: Encrypts outgoing messages and decrypts incoming ones using the established shared key, ensuring confidentiality.

In this paper, the web PKI used in TLS is assumed for public key cryptographic operations. However, other PKIs operated by the Internet community (say, ICANN or regional Internet registries) like RPKI can also be used in the proposed framework.

### 2.2 Securing Messages

The Security Oracle allows existing protocols to evolve into their secure variants by leveraging the above functions. Consider a typical request-response interaction: the initiator sends a `Request message`, and the responder returns a `Response message`. Without additional measures, these exchanges are vulnerable to spoofing, tampering, and man-in-the-middle attacks.

For instance, in Dynamic Host Configuration Protocol (DHCP), a client requests for network configuration (`DHCPDISCOVER`) and a

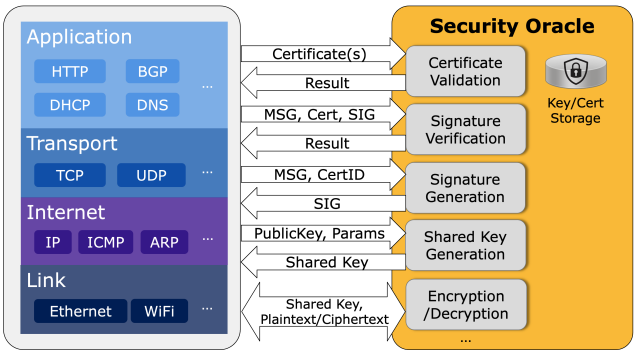


**Figure 1: The Security Oracle operates along with the data plane, which is depicted as a vertical plane. Any protocol (across the layers) in the data plane can request some security services from the Security Oracle, which runs in a holistic fashion. The central element of the Security Oracle is certificates, which are used to (i) attest the ownership of its identity (i.e., its private key) to a counterpart, and (ii) verify the ownership of the identity of a counterpart.**

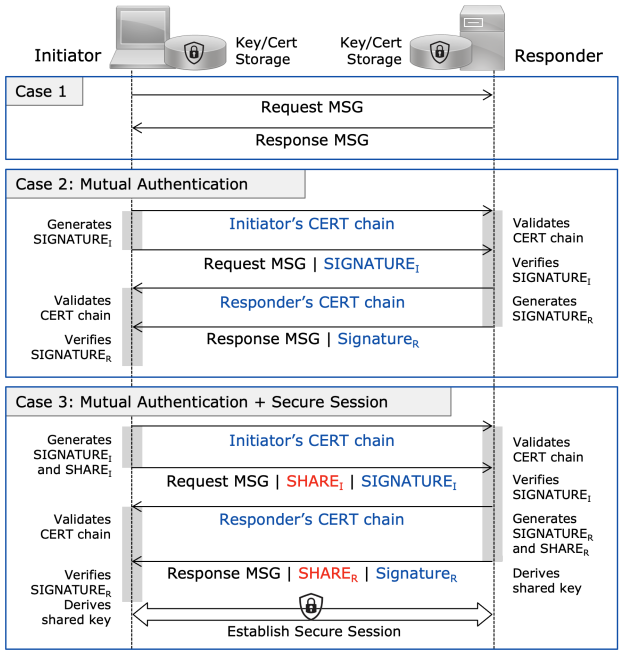


**Figure 2: The message exchange between an initiator and a responder is depicted across three distinct scenarios. When a protocol incorporates security services from the Security Oracle, it can leverage the second and third scenarios.**

server responds with configuration parameters (DHCPOFFER), followed by further exchanges (DHCPREQUEST, DHCPACKNOWLEDGEMENT). Each of these messages is susceptible to unauthorized manipulation.

By incorporating the Security Oracle, protocols can exchange authenticated and integrity-protected messages. As shown in Figure 2 (case 2), the initiator first sends its certificate chain, followed by the signed `Request message`. The responder replies with its certificate chain and a signed `Response message`. Thus, protocols must be extended to carry multiple certificates and signatures, potentially requiring fragmentation and reassembly.

Signature generation and verification rely on the Security Oracle's private keys and trusted CA certificates, respectively. To counter replay attacks, protocols should include timestamps or nonces before hashing a message for signature.

If confidentiality is desired, the protocol instances can also exchange key shares ($SHARE_I$ and $SHARE_R$) (e.g., Diffie-Hellman parameters) to generate a shared key, as shown in Figure 2 (case 3). In this scenario, the Security Oracle not only handles authentication and integrity (in case 2) but also manages shared key derivation and subsequent traffic encryption/decryption.

## 3 Prototype Implementations

In this section, we demonstrate how to extend existing Internet protocols by integrating a Security Oracle. Implemented using the SSL library, the Security Oracle facilitates signature generation/validation and certificate verification. We apply this approach to the ICMP and DHCP protocols and verify successful signature and certificate integration by capturing and analyzing packets with Wireshark.

### 3.1 How to Extend Protocols: A Focus on Size

When attaching signatures and certificate chains to messages, it is crucial to seek to prevent the total data size from exceeding the available payload space, which otherwise causes fragmentation and increases transmission delays (due to more round-trip times (RTTs)). The sizes of signatures and certificates depend on the chosen cryptographic algorithm. We tested five algorithms: RSA with 2048- and 4096-bit keys, and ECDSA with 256-, 384-, and 521-bit keys. ECDSA-256 provides security comparable to RSA-4096 while producing significantly smaller signatures and certificates. As shown in Table 1, this makes ECDSA more efficient, although the exact certificate size may vary depending on the configuration parameters.

***IEEE 802.11.*** To prevent rogue WiFi access points (APs), it is essential to verify the integrity of a beacon frame. An IEEE 802.11 beacon frame includes a 24-byte MAC header containing critical information such as the source address [5]. Its body can be up to 2312 bytes, typically containing around 200 bytes of parameters and extensions. Using ECDSA, we can embed a signature and two certificates (one from the issuing CA and one from the AP) within a single beacon frame. Otherwise, these credentials must be split over multiple frames.

***Internet Control Message Protocol (ICMP).*** ICMP operates at the network layer and thus requires only a 20-byte IP header. An `ICMP ECHO` request/reply is 8 bytes long. Assuming an Ethernet frame payload of up to 1500 bytes, an `ICMP ECHO` message can allocate 1472 bytes for additional data. This space is sufficient to include a signature and two certificates if the ECDSA-256 algorithm is employed.

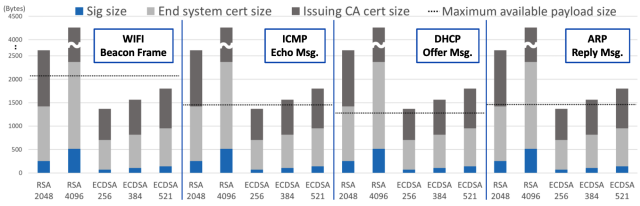


**Figure 3: Depending on the signature algorithms, the sizes of a signature and a certificate chain vary substantially. The maximum available payload of a protocol may not be able to carry such authentication data in a single packet.**

| | RSA-2048 | RSA-4096 | ECDSA-256 | ECDSA-384 | ECDSA-521 |
|---|---|---|---|---|---|
| **Signature size** | 256 bytes | 514 bytes | 71 bytes | 103 bytes | 139 bytes |
| **End system cert. size** | 1164 bytes | 1858 bytes | 631 bytes | 713 bytes | 814 bytes |
| **Issuing CA cert. size** | 1196 bytes | 1891 bytes | 664 bytes | 745 bytes | 847 bytes |
| **Root CA cert. size** | 1159 bytes | 1854 bytes | 623 bytes | 709 bytes | 806 bytes |

**Table 1: Signature and certificate size with various algorithm**

***Dynamic Host Configuration Protocol (DHCP).*** DHCP runs over UDP. A UDP header is 8 bytes long, and an IP header is 20 bytes long, while a typical `DHCPOFFER` message with mandatory options is about 267 bytes long. Approximately 1200 bytes (of an Ethernet frame) remain available for a payload, allowing for inclusion of a signature and two certificates when using ECDSA-256.

***Address Resolution Protocol (ARP).*** An ARP message is 28 bytes and is carried directly in an Ethernet frame without an IP header. With a maximum payload size of about 1478 bytes, it can contain a signature and two certificates only if ECDSA-256 is used.

### 3.2 Security Oracle

Our implementation of the Security Oracle is based on the OpenSSL library [7] for certificate validation and signature operations. The Security Oracle, a suite of security services running in userspace, continuously runs and responds to requests from data-plane protocols operating at various layers, including the kernel, device drivers, and user-level applications. Upon receiving a request for security services—such as signature generation, signature verification, or certificate validation—the Security Oracle performs the necessary operations and creates a corresponding security context. This context, maintained in memory, is then accessible to its client (i.e., initiator or responder), facilitating secure communications.

### 3.3 Extending ICMP and DHCP

To demonstrate the integration of the Security Oracle into existing Internet protocols, we applied our approach to ICMP and DHCP. The ICMP implementation required modifications to the Linux kernel source (i.e., `/net/ipv4/icmp.c`), while the DHCP implementation relied on the userspace code that constructs and processes DHCP messages before they reach or leave the network stack.

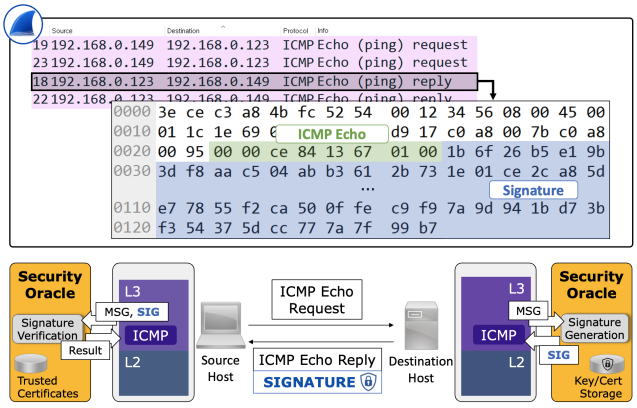


**Figure 4: A signature is appended to the `ICMP Echo Reply` packet.**

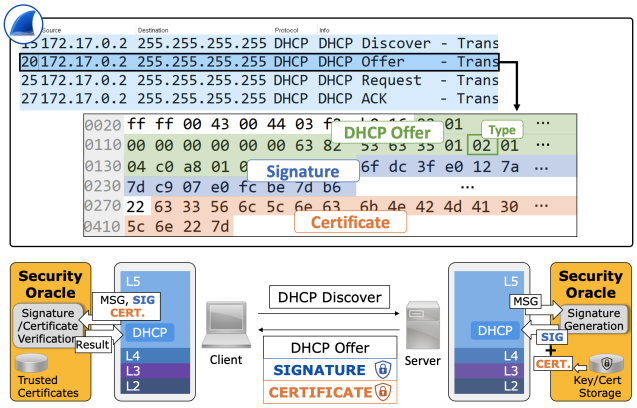


**Figure 5: A signature and a certificate chain are appended to the `DHCPOFFER` message.**

***ICMP Integration***. As illustrated in Figure 4, When an initiator sends an `ICMP Echo Request`, the responder intercepts it and constructs an `ICMP Echo Reply`. During this construction, the Security Oracle generates a signature that is attached to the message. Upon receiving the `ICMP Echo Reply`, the initiator sends the message and signature to the Security Oracle, which verifies the signature's authenticity before allowing the message to proceed through the network stack.

***DHCP Integration***. DHCP is managed entirely in userspace by dedicated client and server applications. We leverage Python-based DHCP client/server modules `DHCPMessage` and `Option90` that allow direct manipulation of DHCP messages at the application layer. As depicted in Figure 5, The client begins by transmitting a `DHCPDISCOVER` message to the server. Upon receiving this message, the server constructs a `DHCPOFFER` message. The Security Oracle then signs the message and appends the signature and the corresponding certificate chain. Once the client receives the `DHCPOFFER`, it forwards the message and certificates to the Security Oracle, which validates the chain and verifies the signature.

These extensions to ICMP and DHCP demonstrate the practical feasibility of integrating the Security Oracle's services into existing Internet protocols, ensuring enhanced authenticity and integrity while maintaining compatibility with standard network operations.

## 4 Conclusion

In this paper, we introduced the Security Oracle, which is a unified framework that provides security services—authentication, integrity, and confidentiality—to existing network protocols. The proposed approach addresses the limitations of individual and uncoordinated network security solutions, each of which targets specific protocols or networks. The Security Oracle allows a given protocol to enhance its security by providing APIs such as certificate validation, signature generation/verification, and key share exchange.

Through carefully designed case studies and prototype experiments, we validated the practical feasibility of the proposed Security Oracle. By comparing signature and certificate sizes against the maximum payload sizes of existing protocols, we showed the viability of the Security Oracle. Using a standard SSL library, we successfully implemented the Security Oracle by extending `ICMP Echo` and `DHCPOFFER` messages, highlighting its technical effectiveness and operational viability.

## Acknowledgments

This work was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2025-2021-0-02048) supervised by the IITP (Institute for Information Communications Technology Planning Evaluation), and also supported by IITP under Next-generation Cloud-native Cellular Network Leadership Program(IITP-2025-RS-2024-00418784) grant funded by the Korea government (MSIT), and also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2022R1A2C2011221, No. RS-2023-00220985).

## References

[1] Patrick Kwadwo Agyapong and Marvin Sirbu. 2012. Economic incentives in information- centric networking: implications for protocol design and public policy. *IEEE Communications Magazine* 50, 12 (2012), 18–26. https://doi.org/10.1109/MCOM.2012.6384447

[2] Ashok Anand, Fahad Dogar, Dongsu Han, Boyan Li, Hyeontaek Lim, Michel Machado, Wenfei Wu, Aditya Akella, David G. Andersen, John W. Byers, Srinivasan Seshan, and Peter Steenkiste. 2011. XIA: An Architecture for an Evolvable and Trustworthy Internet. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks* (Cambridge, Massachusetts) *(HotNets-X)*. Association for Computing Machinery, New York, NY, USA, Article 2, 6 pages. https://doi.org/10.1145/2070562.2070564

[3] Taejoong Chung, Roland Van Rijswijk-Deij, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. 2017. A Longitudinal, End-to-End View of the DNSSEC Ecosystem. In *Proceedings of the 26th USENIX Conference on Security Symposium* (Vancouver, BC, Canada) *(SEC'17)*. USENIX Association, USA, 1307–1322.

[4] Paul E. Hoffman. 2023. *DNS Security Extensions (DNSSEC).* RFC 9364. IETF.

[5] IEEE. 2009. *IEEE Std 802.11n™-2009 (Amendment to IEEE Std 802.11-2007), Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Higher Throughput.* Technical Report.

[6] Matt Lepinski and Stephen Kent. 2012. An Infrastructure to Support Secure Internet Routing. RFC 6480. https://doi.org/10.17487/RFC6480

[7] OpenSSL Team. 2021. OpenSSL 1.1.1t. [Software]. https://www.openssl.org/.

[8] Adrian Perrig, Pawel Szalachowski, Raphael M. Reischuk, and Laurent Chuat. 2017. *SCION: A Secure Internet Architecture.* Springer International Publishing.

[9] Eric Rescorla. 2018. *The Transport Layer Security (TLS) Protocol Version 1.3.* RFC 8446. IETF.

[10] Karen Seo and Stephen Kent. 2005. *Security Architecture for the Internet Protocol.* RFC 4301. IETF.