HOKKAIDO UNIVERSITY

Title	Towards Privacy Enhanced NS Record-Based Key Distribution for Encrypted SNI
Author(s)	Sunahara, Satoru; Kuroda, Niko; Hagihara, Shigeki et al.
Citation	2025 30th Asia-Pacific Conference on Communications (APCC), 1-2 https://doi.org/10.23919/apcc64555.2025.11279839
Issue Date	2025-11-26
Doc URL	https://hdl.handle.net/2115/100415
Rights	© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Type	conference paper
File Information	11279839.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

Towards Privacy Enhanced NS Record-Based Key Distribution for Encrypted SNI

Satoru Sunahara
Hokkaido University
Sapporo, Hokkaido, Japan
suna@iic.hokudai.ac.jp

Niko Kuroda
Chitose Institute of Science
Chitose, Hokkaido, Japan
b2220920@photon.chitose.ac.jp

Shigeki Hagihara
Chitose Institute of Science
Chitose, Hokkaido, Japan
s-hagiha@photon.chitose.ac.jp

Yong Jin
Institute of Science Tokyo
Meguro, Tokyo, Japan
yongj@cii.isct.ac.jp

Katsuyoshi Iida
Hokkaido University
Sapporo, Hokkaido, Japan
iida@iic.hokudai.ac.jp

Nariyoshi Yamai
Tokyo University of Agriculture
Koganei, Tokyo, Japan
nyamai@cc.tuat.ac.jp

***Abstract*—While payload encryption for privacy protection between full-service resolvers and authoritative DNS servers has been advancing, privacy-sensitive information in communication headers, which remain unencrypted under conventional TLS, remains unaddressed. If the draft-ietf-tls-esni-25 is applied directly in this scenario, the SNI, which is privacy-sensitive information, is exposed during the process of obtaining the public key for encrypting it. Our contribution lies in proposing a novel public key distribution mechanism designed to encrypt privacy-related information contained in communication headers between full-service resolvers and authoritative DNS servers, thereby providing a direction for enhancing privacy protection.**

***Index Terms*—privacy preservation, DNS, key distribution, Encrypted SNI.**

I. Introduction

In conventional Transport Layer Security (TLS) communication, the confidentiality of sensitive information contained in the payload has been enhanced through encryption, thereby strengthening privacy protection. However, in recent years, privacy concerns have emerged due to threats such as censorship and blocking based on inspection of unencrypted elements in the TLS handshake, such as the Server Name Indication (SNI), as well as user profiling through traffic analysis of communication headers [1] [2]. To address these challenges, the Internet Engineering Task Force (IETF) is currently discussing methods for protecting SNI information within TLS communications (draft-ietf-tls-esni-25) [3]. Furthermore, the IETF is also discussing encryption mechanisms to protect privacy-related information contained in DNS communications, and has recently standardized methods for encrypting communication between full-service resolvers and authoritative DNS servers (RFC 9539).

However, when applying the SNI protection mechanism described in draft-ietf-tls-esni-25 to encrypted communication between full-service resolvers and authoritative DNS servers, a fundamental challenge arises. The Encrypted Client Hello (ECH) mechanism assumes that the encryption public key (ECHConfig) is pre-obtained via an HTTPS DNS record.

To protect the SNI sent from a resolver to an authoritative server, the resolver must first query the server's HTTPS record to retrieve its public key. However, this exchange occurs before the ServerHello can be encrypted, leaving the key retrieval communication unprotected and potentially exposing privacy-sensitive information.

To address this, the resolver must obtain the target authoritative server's public key before initiating communication. The proposed solution to this problem involves embedding the public key within the NS (Name Server) record. Dempsky achieved the encryption of DNS queries by registering a public key information in the NS record [4]. We propose a mechanism for registering key information in NS records to protect the Server Name Indication (SNI) between full-service resolvers and authoritative DNS servers through encryption, addressing the unresolved issue in draft-ietf-tls-esni-25 and RFC 9539.

Our approach uses DNS NS records for this purpose, leveraging the delegation information they provide from the parent authoritative DNS server.

II. Proposed Method

We propose a method to obtain the public key used for encrypting the SNI in the ECH during TLS communication between a full-service resolver and an authoritative DNS server by embedding it in the NS record. Figure 1 compares the conventional draft-ietf-tls-esni-25 method with our approach. In the draft method, the server's public key is provided via an SVCB or HTTPS record. Until this key is obtained, the SNI in the ServerHello remains unencrypted.

In our approach, we assume the root authoritative DNS server's public key is included in the NS records listed in the root.hints file on the full-service resolver's host. Based on this, the resolver and the root authoritative DNS server can establish a shared key using Hybrid Public Key Encryption (HPKE) as standardized in RFC 9180. This enables the resolver to encrypt the SNI from its first communication with the root authoritative DNS server.

The `.com` authoritative DNS server similarly pre-registers its public key in the NS record of the root authoritative DNS server. Thus, when the resolver receives the NS record from the root server, it obtains both the name server's name and public key. This process is recursively repeated for delegated DNS servers.

Our method allows SNI encryption in the ServerHello from the initial communication, eliminating the SVCB/HTTPS key retrieval step of the draft method, thereby reducing latency. Upon receiving a public key, the resolver generates a shared key per RFC 9180 and encrypts privacy-sensitive header information previously unencrypted in TLS. The resolver sends the encrypted message along with its public key to the authoritative DNS server, which generates the shared key and decrypts the headers.

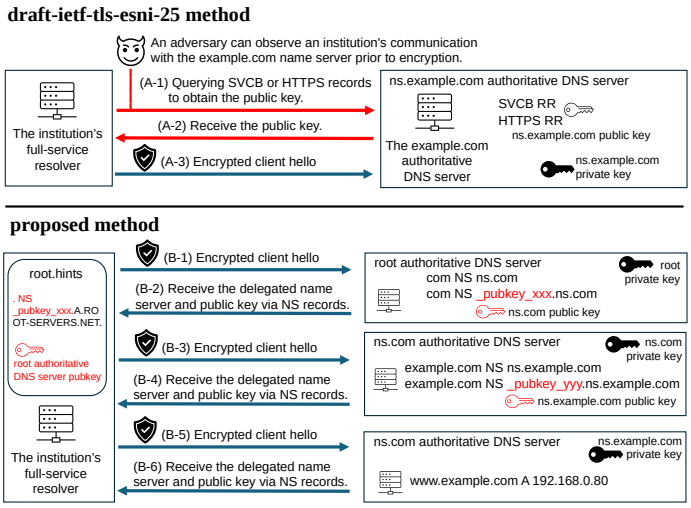

Fig. 1. Differences in key exchange between the draft-ietf-tls-esni-25 method and the proposed approach.

III. Prototype Implementation

To demonstrate the feasibility of our approach, we implemented in Python a proxy for a full-service resolver that retrieves authoritative DNS server public keys from the root hints file and NS records, and a proxy for an authoritative DNS server that receives the resolver's public key and encrypted headers, then performs decryption. For the verification process, BIND 9 was utilized as the DNS server software.

As our method requires shared key generation based on RFC 9180, the NS record must store a public key of sufficient cryptographic strength in domain name form. Following the National Institute of Standards and Technology (NIST) recommendations, we use Curve P-384, projected to remain secure beyond 2031. The Curve P-384 public key in the NS record must satisfy the 255-octet domain name limit and the 63-octet label limit in RFC 1035, and conform to the alphabet, digit, hyphen, and period restrictions in RFC 952.

We addressed these constraints by representing the public key in binary form using Distinguished Encoding Rules (DER) and applying base62 encoding. Base62 was chosen over base64 (RFC 4648) because base64 includes characters not permitted by RFC 952, whereas base62 uses only alphanumeric characters (A–Z, a–z, 0–9).

Figure 2 shows an example in which the public keys encoded using base62 are registered in the NS records of the authoritative DNS servers for the top-level domain.

```
example.com NS ns1.example.com.
example.com NS ns2.example.com.

example.com NS
_pubkey_3yLK39JW61oloBURhVZi9notZquixM7iYz1lrjzjsz0L0LYDFIpvpNc.0a5qn4qvsGfJe3aK5ozCfP7W
K0vrTKHm1EokM3xjBR.ns1.example.com._pubkey_3yLK39JW61oloBURhVZi9notZquixM7iXBbSy3Fl53l1
8ISlw9sZ4Q.FHV0s3TQe6Xm7G5tvexeuNGSApyVtopQvKNHJ4trag7.ns2.example.com.

ns1.example.com A      192.168.53.10
ns2.example.com A 192.168.53.20

_pubkey_3yLK39JW61oloBURhVZi9notZquixM7iYz1lrjzjsz0L0LYDFIpvpNc.0a5qn4qvsGfJe3aK5ozCfP7W
K0vrTKHm1EokM3xjBR.ns1.example.com._pubkey_3yLK39JW61oloBURhVZi9notZquixM7iXBbSy3Fl53l1
8ISlw9sZ4Q.FHV0s3TQe6Xm7G5tvexeuNGSApyVtopQvKNHJ4trag7.ns2.example.com. A   192.168.53.10

_pubkey_3yLK39JW61oloBURhVZi9notZquixM7iYz1lrjzjsz0L0LYDFIpvpNc.0a5qn4qvsGfJe3aK5ozCfP7W
K0vrTKHm1EokM3xjBR.ns1.example.com._pubkey_3yLK39JW61oloBURhVZi9notZquixM7iXBbSy3Fl53l1
8ISlw9sZ4Q.FHV0s3TQe6Xm7G5tvexeuNGSApyVtopQvKNHJ4trag7.ns2.example.com. A 192.168.53.20
```

Fig. 2. An example of an NS record applying the proposed method.

In the authoritative DNS servers for the `.com` domain, `ns1.example.com` and `ns2.example.com` are designated as the delegation targets for `example.com`, and public keys prefixed with `_pubkey_` can be registered in NS records. The public key for `ns1.example.com` is placed between the `_pubkey_` prefix and `ns1.example.com.` If the key exceeds the label length limit, periods are inserted to continue it.

Currently, underscores in NS records are not permitted under existing standards. In BIND9, zone files containing such records cannot be loaded unless the check-names ignore option is explicitly set in named.conf. Our proposal intentionally uses an underscore-containing prefix to indicate that an NS record includes a public key, despite the lack of standardization.

The full-service resolver retrieves the authoritative server's public key from the NS record, derives an AES shared secret via key exchange, and encrypts communication. It then sends its own public key with the encrypted data to the authoritative server, which derives the AES key from the received public key and decrypts the data.

IV. Research Plan

In the future, we plan to verify the applicability of our proposed approach to privacy protection mechanisms such as the encryption of ECH included in the TLS ServerHello message and proxy-based schemes like ODoH (RFC 9230). We also intend to pursue a formal verification of the security properties of our method.

References

[1] Z. Chai, A. Ghafari, and A. Houmansadr, "On the importance of Encrypted-SNI (ESNI) to censorship circumvention," in *9th USENIX Workshop on Free and Open Communications on the Internet (FOCI 19)*. Santa Clara, CA: USENIX Association, Aug. 2019.

[2] N. Niere, F. Lange, R. Merget, and J. Somorovsky, "Transport layer obscurity: Circumventing sni censorship on the tls-layer," in *2025 IEEE Symposium on Security and Privacy (SP)*, 2025, pp. 1344–1362.

[3] E. Rescorla, K. Oku, N. Sullivan, and C. A. Wood, "TLS Encrypted Client Hello," Internet Engineering Task Force, Internet-Draft draft-ietf-tls-esni-25, Jun. 2025, work in Progress.

[4] D. J. Bernstein, "Dnscurve: usable security for dns," https://dnscurve.org/. (2009).