Artifacts Evaluated acm Functional V1.1

Artifacts Evaluated acm Reusable V1.1

Delegation of TLS Authentication to CDNs using Revocable Delegated Credentials

Daegeun Yoon, Taejoong Chung, Yongdae Kim

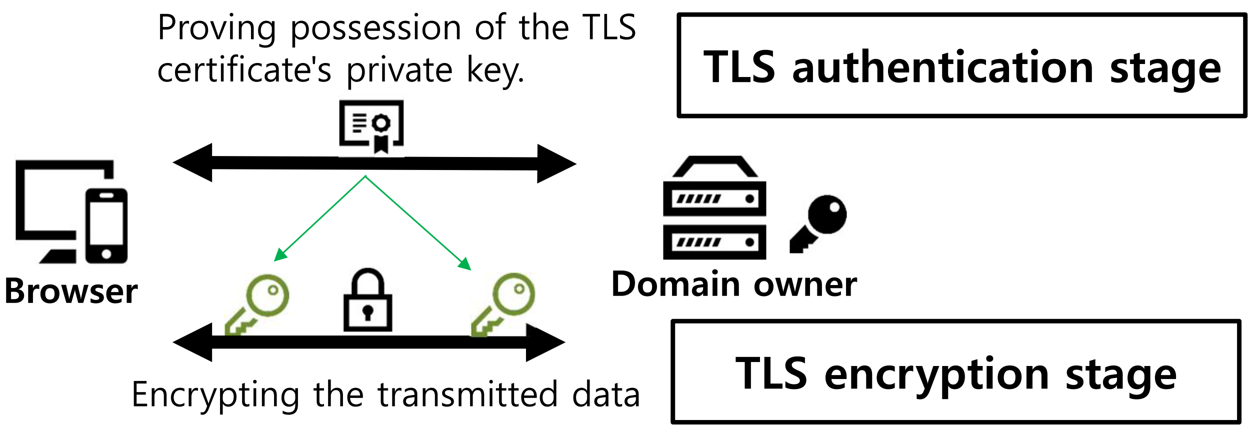
ETRI Electronics and Telecommunications Research Institute

VT VIRGINIA TECH.

KAIST SysSec Lab. KAIST

TLS Protocol

- A TLS protocol consists of two stages: authentication and encryption.
  - TLS authentication: proving the domain owner’s identity to a browser
  - TLS encryption: encrypting the transmitted data



Proving possession of the TLS certificate's private key.

TLS authentication stage

Browser

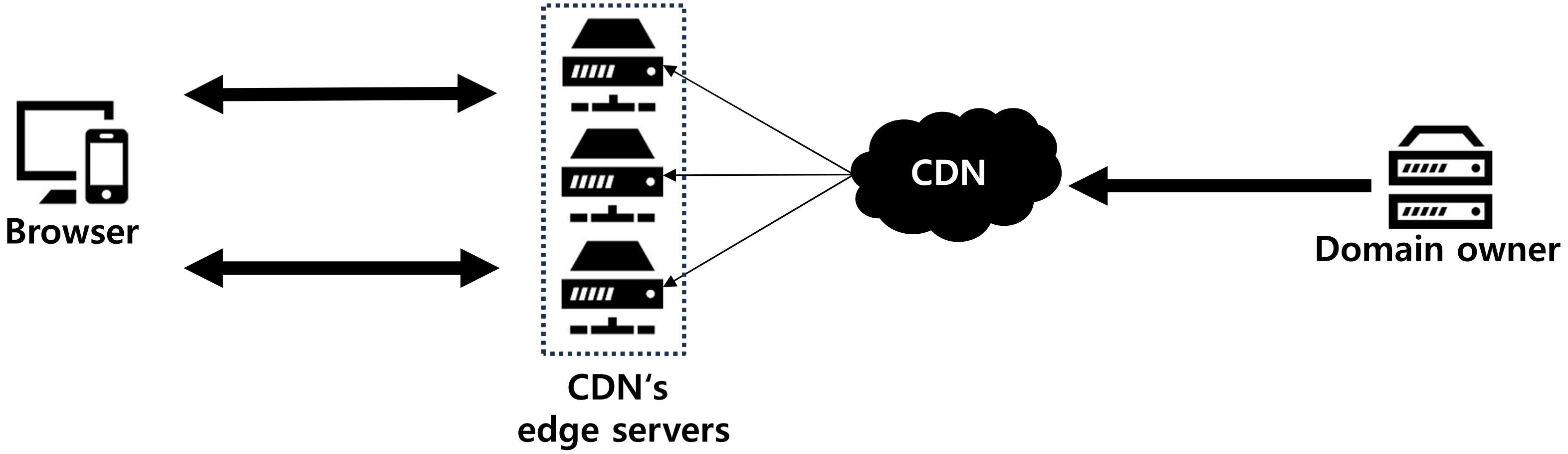
Domain owner

Encrypting the transmitted data

TLS encryption stage

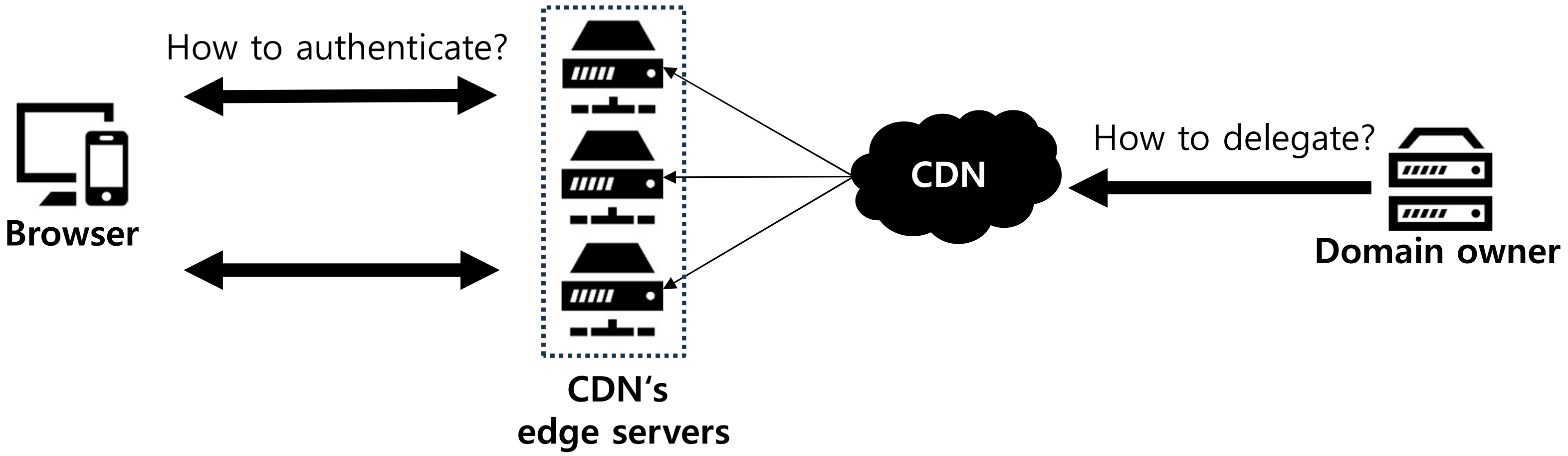
Delegation of TLS Authentication to CDNs

- Today, numerous web communications rely on intermediaries (e.g., CDNs).
  - Domain owners need to delegate TLS authentication to CDNs.



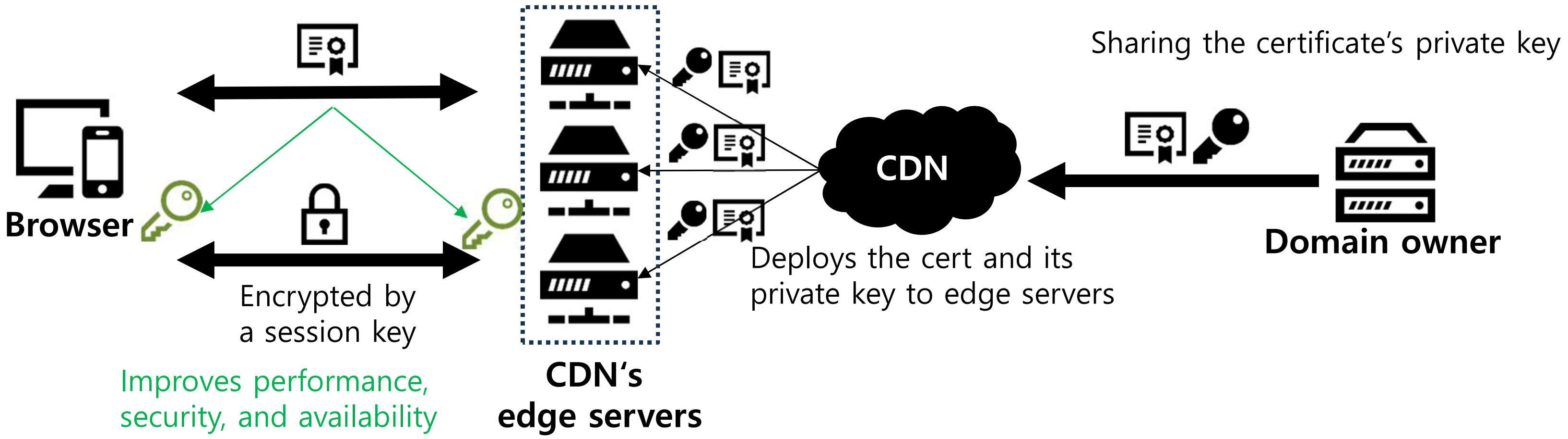
Delegation of TLS Authentication to CDNs

- ❖ Today, numerous web communications rely on intermediaries (e.g., CDNs).
  - – Domain owners need to delegate TLS authentication to CDNs.
- ❖ However, the TLS standard does not support this communication model.



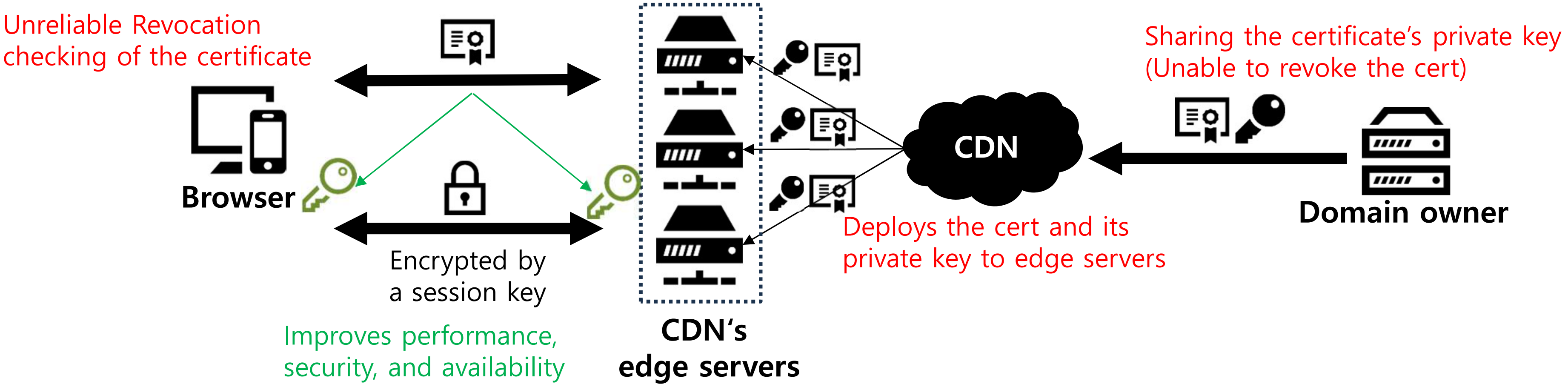
Delegation of TLS Authentication to CDNs

- Today, numerous web communications rely on intermediaries (e.g., CDNs).
  - Domain owners need to delegate TLS authentication to CDNs.
- However, the TLS standard does not support this communication model.
  - Sharing the certificate's private key is a common method for delegation.
    - CDNs Generate the certificate and its private key.
    - Domain owners upload their certificate and its private key.



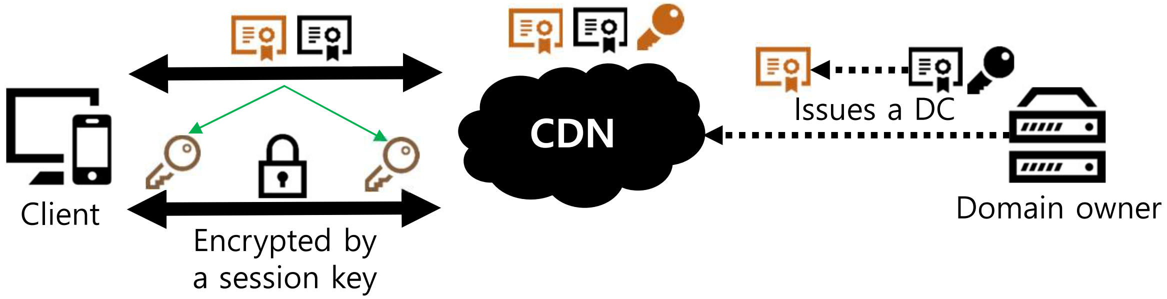
Delegation of TLS Authentication to CDNs

- Today, numerous web communications rely on intermediaries (e.g., CDNs).
  - Domain owners need to delegate TLS authentication to CDNs.
- However, the TLS standard does not support this communication model.
  - Sharing the certificate's private key is a common method for delegation.
    - CDNs Generate the certificate and its private key.
    - Domain owners upload their certificate and its private key.


Unreliable Revocation
checking of the certificate
Browser
Encrypted by
a session key
Improves performance,
security, and availability
CDN's
edge servers
CDN
Deploys the cert and its
private key to edge servers
Sharing the certificate's private key
(Unable to revoke the cert)
Domain owner

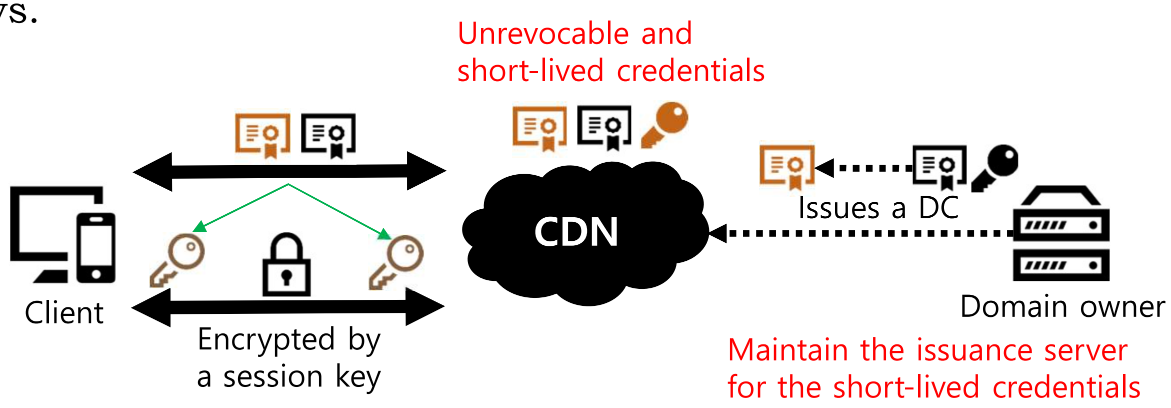
Existing Solutions: Delegated Credential

- RFC 9345 defines Delegated Credentials (DCs).
  - Domain owners issue DCs to CDNs for TLS authentication.
  - CDNs perform TLS authentication using the DCs and their private keys.
- DCs do not provide a method of distributing their revocation status.
  - Even if a DC is compromised, the domain owner cannot revoke the DC.
  - Inevitably, DCs are designed to be short-lived (at most 7 days).
    - Domain owners require an issuance server capable of issuing DCs to CDNs every 7 days.



Existing Solutions: Delegated Credential

- RFC 9345 defines Delegated Credentials (DCs).
  - Domain owners issue DCs to CDNs for TLS authentication.
  - CDNs perform TLS authentication using the DCs and their private keys.
- DCs do not provide a method of distributing their revocation status.
  - Even if a DC is compromised, the domain owner cannot revoke the DC.
  - Inevitably, DCs are designed to be short-lived (at most 7 days).
    - Domain owners require an issuance server capable of issuing DCs to CDNs every 7 days.


Unrevocable and short-lived credentials

Maintain the issuance server for the short-lived credentials

Design Goals

- We design **Revocable** Delegated Credentials (RDCs) that satisfy the five goals to achieve secure delegation of TLS authentication

Design Goals

- We design **Revocable** Delegated Credentials (RDCs) that satisfy the five goals to achieve secure delegation of TLS authentication

- No sharing of the domain owner's private key

Design Goals

- We design **Revocable** Delegated Credentials (RDCs) that satisfy the five goals to achieve secure delegation of TLS authentication

- No sharing of the domain owner’s private key
- Revoking the delegation key without revoking the TLS certificate

Design Goals

- We design **Revocable** Delegated Credentials (RDCs) that satisfy the five goals to achieve secure delegation of TLS authentication

- No sharing of the domain owner's private key
- Revoking the delegation key without revoking the TLS certificate
- Retaining control of revoking delegation keys

Design Goals

- We design **Revocable** Delegated Credentials (RDCs) that satisfy the five goals to achieve secure delegation of TLS authentication

- No sharing of the domain owner’s private key
- Revoking the delegation key without revoking the TLS certificate
- Retaining control of revoking delegation keys
- Compliance of RDC with the current standards and infrastructure

Design Goals

- We design **Revocable** Delegated Credentials (RDCs) that satisfy the five goals to achieve secure delegation of TLS authentication

- No sharing of the domain owner's private key
- Revoking the delegation key without revoking the TLS certificate
- Retaining control of revoking delegation keys
- Compliance of RDC with the current standards and infrastructure
- Retaining benefits of using a CDN

Question

- How can we distribute the revocation status of RDCs?

Question

- How can we distribute the revocation status of RDCs?
- DNS!

Question

- How can we distribute the revocation status of RDCs?
- DNS!

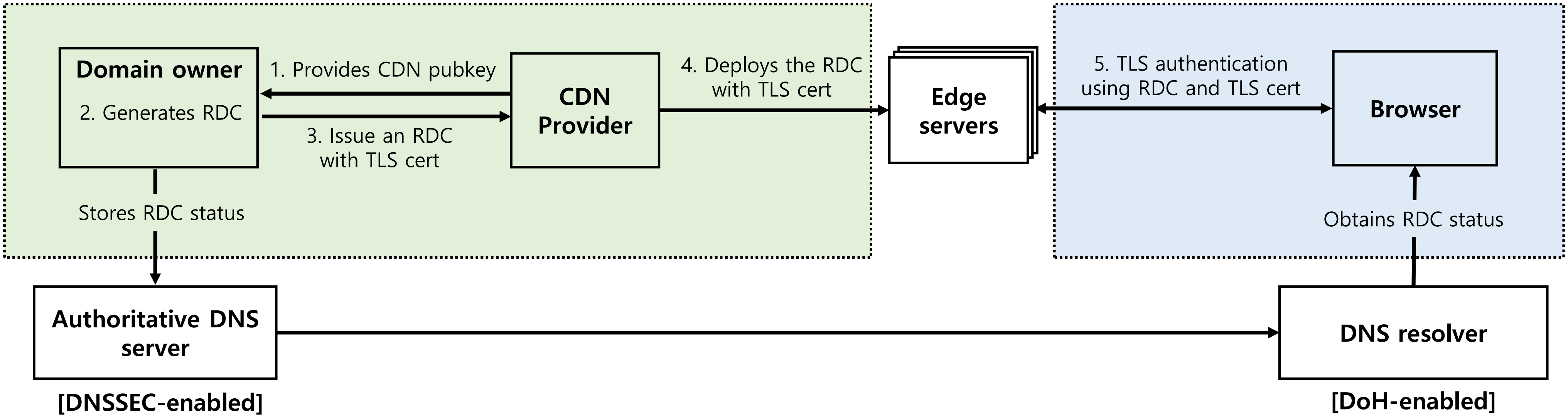
- DNS is an essential component of web communication
  - Not only provide IP addresses, but also provide various types of information for web communication
    - Already support to deliver TLS-level information such as TLSA, SVCB

Question

- How can we distribute the revocation status of RDCs?
- DNS!

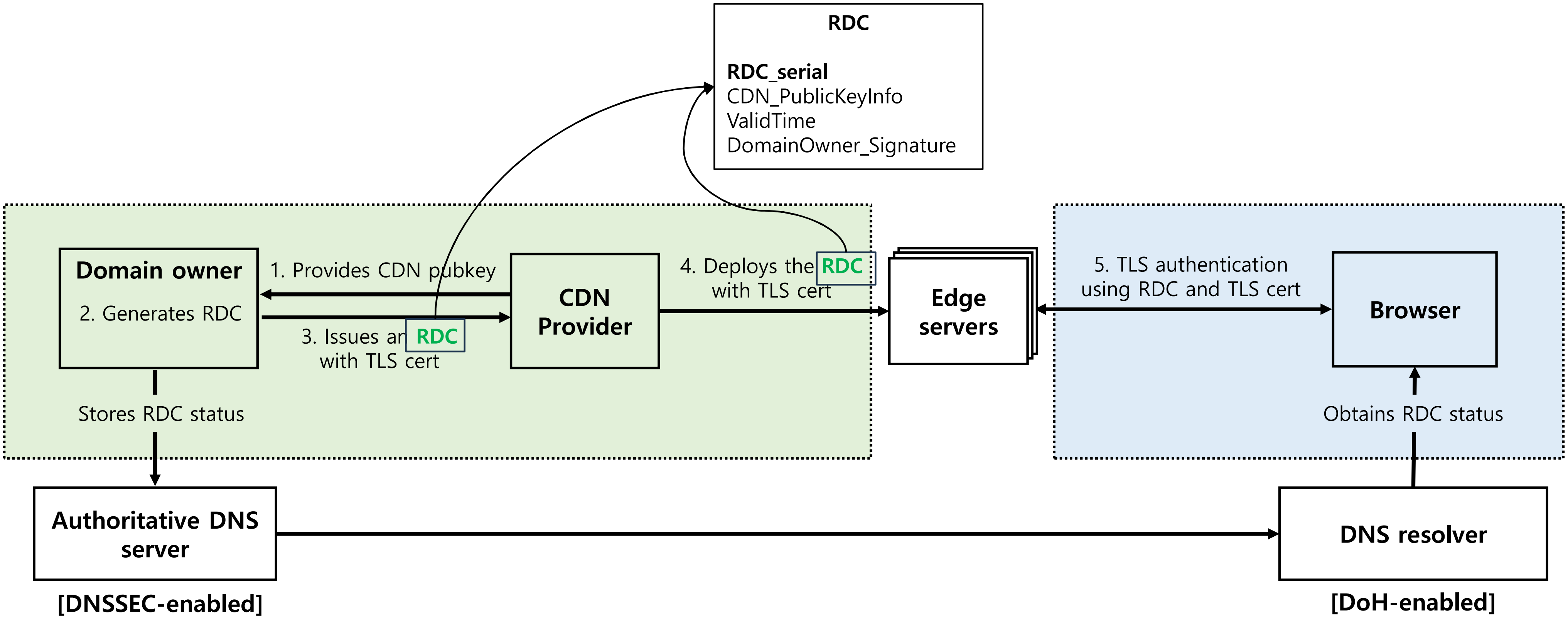
- DNS is an essential component of web communication
  - Not only provide IP addresses, but also provide various types of information for web communication
    - Already support to deliver TLS-level information such as TLSA, SVCB
  - Support security mechanism
    - Integrity: DNSSEC
    - Confidentiality: DoH

Design Overview



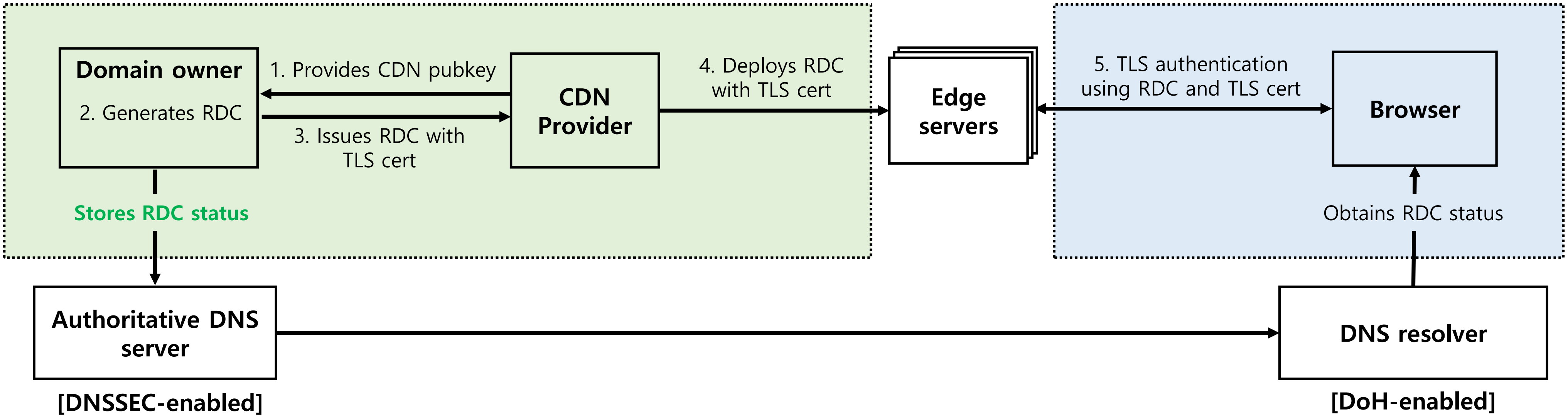
Properties of RDC

- ❖ RDC has a unique identifier, called an “RDC_serial”



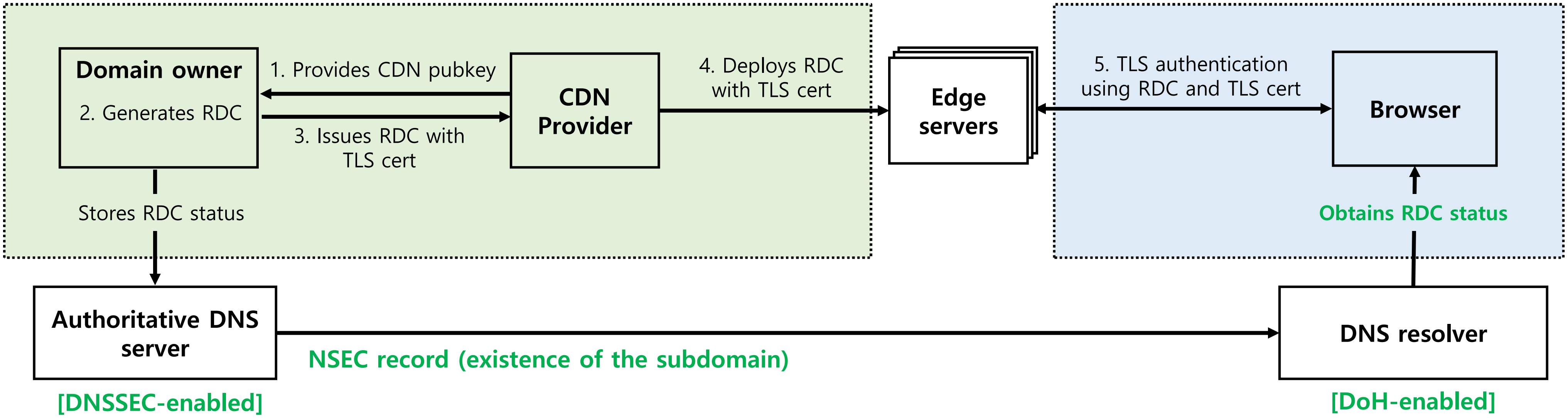
Determination of Revocation Status

- The revocation status of an RDC is determined by existence of the subdomain named <RDC_serial>
  - Revoked if <RDC_serial>.<domain name> exists
  - Valid if <RDC_serial>.<domain name> does not exists



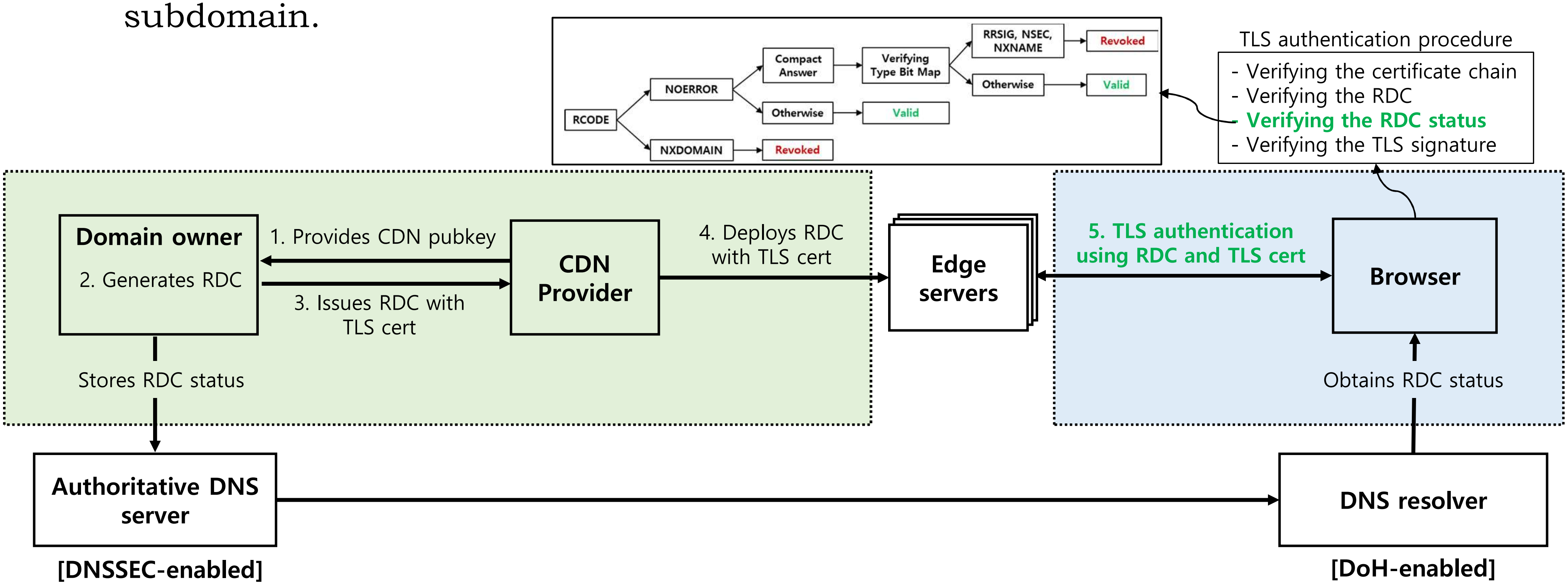
Distribution of Revocation Status

- ❖ Integrity of the RDC revocation status is guaranteed by DNSSEC.
  - – NSEC record, which is a type of DNSSEC record, provides the proof of existence of the domain.
- ❖ Confidentiality of the RDC revocation status is guaranteed by DoH.



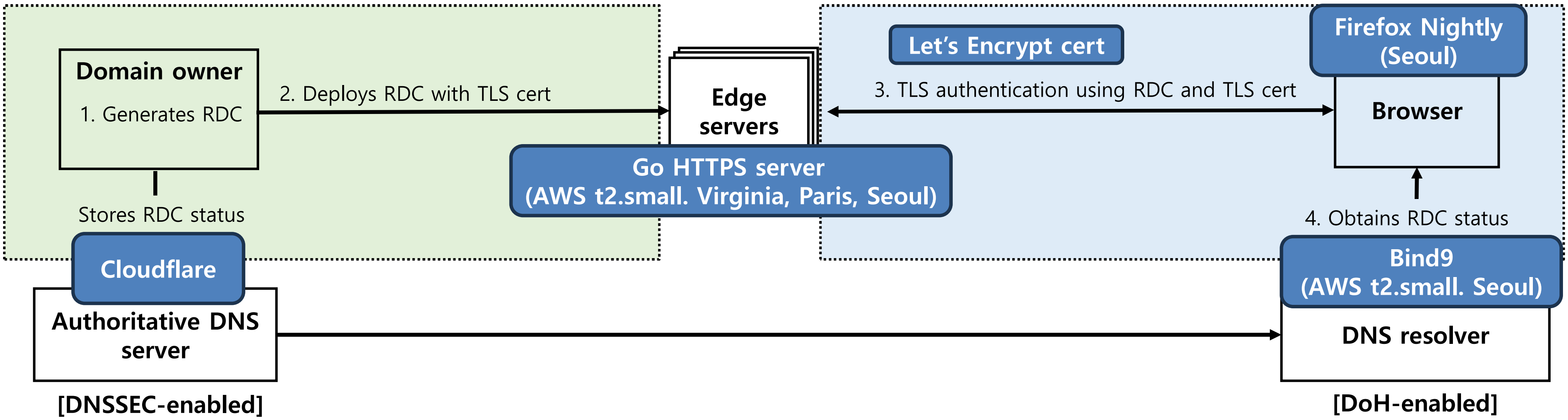
Verification of Revocation Status

- ❖ Browsers obtain the RDC status during the TLS authentication procedure.
  - – Verify the DNS response including NSEC record to determine the existence of the subdomain.



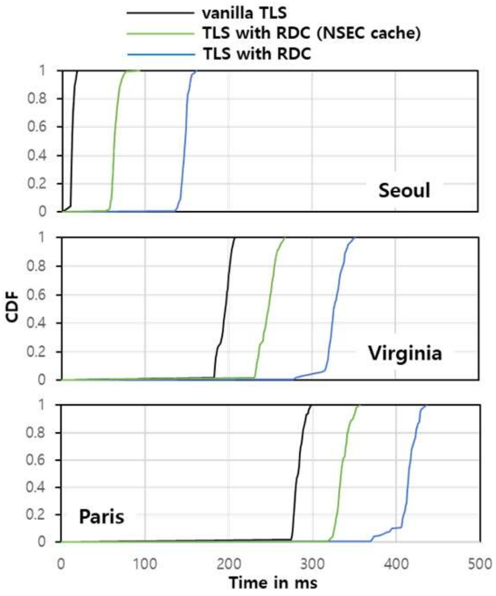
Implementation and Experimental Setup

- Implementing RDC into the Go tls package and the NSS library
  - The Go tls package for the RDC-supporting HTTPS server
  - The NSS library for the RDC-supporting Firefox Nightly browser

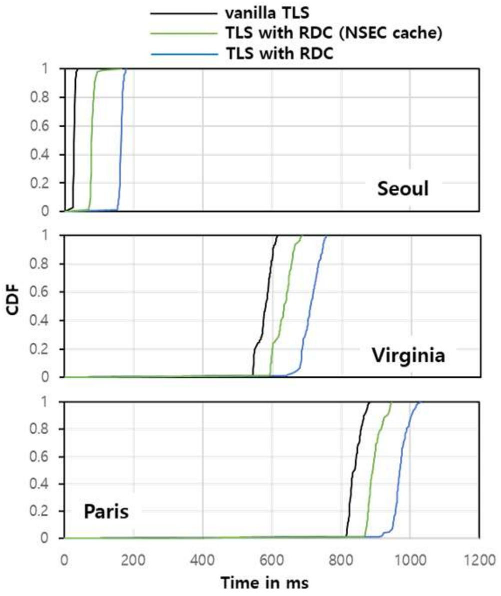


Evaluation

- ❖ Only one-time delay (50-130 ms) compared to the vanilla TLS
  - – Moderate security but better performance than other TLS encryption solutions that introduce overhead for every communication

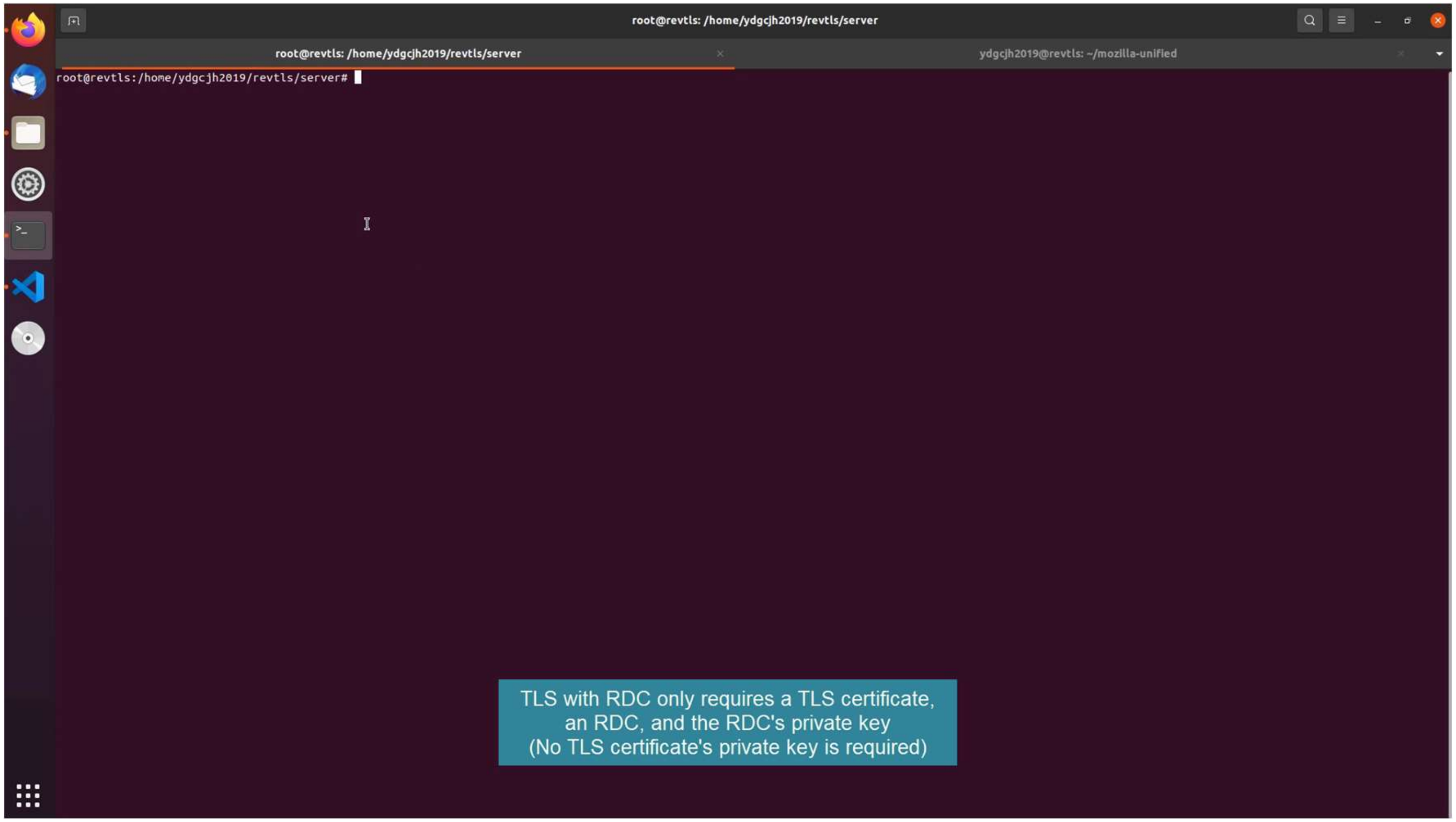


<TLS setup time>



<Page load time>

Demo for Function Evaluation


root@revtls: /home/ydgcjh2019/revtls/server
root@revtls: /home/ydgcjh2019/revtls/server
ydgcjh2019@revtls: ~/mozilla-unified

```
root@revtls:/home/ydgcjh2019/revtls/server# 
```

TLS with RDC only requires a TLS certificate, an RDC, and the RDC's private key (No TLS certificate's private key is required)

Conclusion

- We introduce **Revocable** Delegated Credential (RDC).
  - Leveraging DNS to store and distribute the revocation status
  - Revoking the delegation key without revoking the TLS certificate
  - Retaining control of revoking delegation keys
  - Compliance with the current standards and infrastructure

- We integrated RDC into Go TLS package and the NSS library
  - Enabling RDC support for both HTTPS servers and browsers
  - Validation of an RDC’s revocation status is only associated with a negligible one-time delay.
  - Code available at https://github.com/revtls/revtls

- RDC allows moderate security but better performance with full benefits of CDNs

Thank you!

Daegeun Yoon

dayoon@etri.re.kr (ydgcjh2019@gmail.com)

ETRI
Electronics and Telecommunications Research Institute

VT
VIRGINIA TECH

KAIST

SysSec Lab.
KAIST

Previous Research

- TEE solutions
  - Phoenix [1], Styx [2]
- TLS extension
  - maTLS [3], mcTLS [4]
- DANE solution
  - InviCloak [5]
- Crypto Solution
  - BlindBox [6], Embark [7]

- Most studies focus on protecting the TLS encryption layer.
  - Better security but high trade-offs
    - Performance degradation, inability to use full functionalities of CDNs, additional deployment

[1] Herwig et.al., Usenix Seccurity'20 [2] Wei et.al., IEEE SoC'17 [3] Lee et.al., NDSS'19 [4] Naylor et.al., ACM SIGCOMM'15
[5] Lin et.al., ACM CCS'22 [6] Sherry et.al., ACM SIGCOMM'15 [7] Lan et.al., Usenix NSDI'16