Protecting Users' Sensitive Data From a Third-Party CDN

Shihan Lin and Xiaowei Yang
shihan.lin@duke.edu, xwy@cs.duke.edu
Department of Computer Science, Duke University

1 Motivation

A website which employs a third-party Content Delivery Network (CDN) service faces the following dilemma:

- CDN keeps private keys of its customers' certificates so it can observe all traffic between users and its customers.
- If clients transfer sensitive data directly to the server, the server's IP address is exposed. It is at the risk of Distributed Denial of Service (DDoS) attacks.

26 of Alexa top 100 sites has purchased third-party CDN service:

- 13 sites transfer users' passwords to CDN.
- 13 sites expose their IP addresses in the login procedure.

2 Related Work

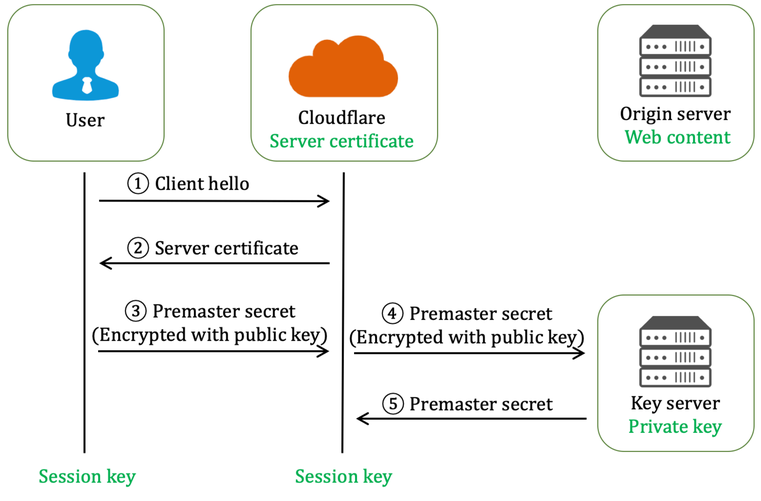

Figure 1: Cloudflare Keyless SSL proposed in 2014

- Session keys of SSL/TLS are still exposed to CDN.
- An open problem proposed in 2016*: Eliminating CDNs' need to have session keys.

*Cangialosi F., etc., Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem. CCS'16

3 Design

Goals

1. Protecting users' sensitive data from leakage.
2. Protecting the server from DDoS attacks.
3. Easy to deploy. No need to modify CDN service and browser.
4. Compatible with HTTPS. No need to modify HTTPS.

Main idea
Add an additional layer of encryption to the client's request over HTTPS and transmit it to the server via CDN.

- CDN cannot decrypt the content. Goal ① ✓
- The server is kept hidden behind CDN. Goal ② ✓

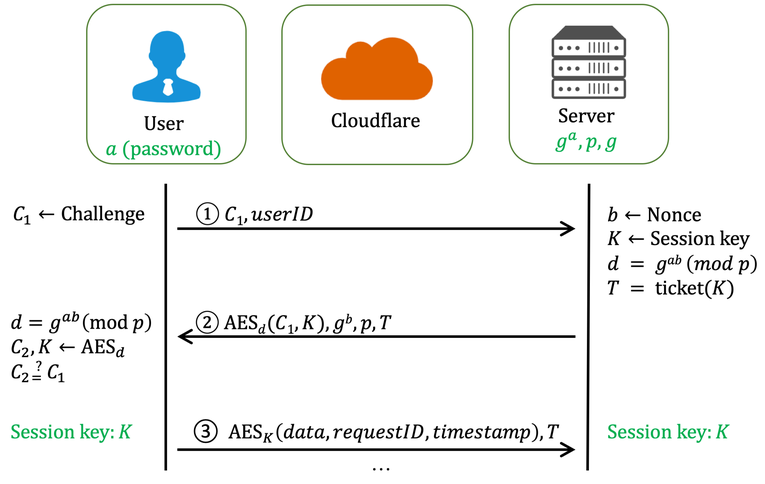

Figure 2: Protocol Design

Key Observation

The client and server have already shared a common secret, i.e., the user's password.

Cloudflare Keyless SSL is employed to remove CDN's access to the private key. The user's password is used as a in Diffie-Hellman algorithm and the server keeps g^a, g, p.

1. Client initiates the client hello request with a random challenge number C_1 and user ID.
2. Server retrieves g^a by user ID and generates a nonce b to calculate g^{ab}. Server generates the session key K and the TLS session ticket T. C_1 and K are encrypted by g^{ab} and returned to the client with g^b, p and T.
3. With the password a, client calculates g^{ab} and obtains the session key K, which is used for future communication.

The protocol runs over stateless HTTPS, so the TLS session ticket is required by the server to decrypt messages across HTTPS connections during the session.

4 Analysis

Security

- User authentication: Only the user knows a.
- Server authentication: Only the server knows g^a, preventing the man-in-the-middle attack
- Prevent replay attacks by timestamp and request ID.
- Forward security even g^a is leaked.
- CDN plays as a shield against DDoS attacks.

Implementation and deployment

- Keyless SSL has been deployed on CDN. Goal ③ ✓
- Totally JavaScript implementable. Goal ③ ✓
- Run over HTTPS. Goal ④ ✓
- Websites' front-end modification is required.

5 Limitation

- Rely on a trust-on-first-use (TOFU) mechanism: Account registration should be secure.
- More computation for each communication with sensitive data.

Duke UNIVERSITY