The Potential Harm of Email Delivery: Investigating the HTTPS Configurations of Webmail Services

Ruixuan Li, Zhenyong Zhang, Jun Shao, *Senior Member, IEEE,* Rongxing Lu, *Fellow, IEEE,* Xiaoqi Jia and Guiyi Wei

Abstract—Webmail, protected by the HTTPS protocol, only works correctly if both the server and client implement HTTPS-related features without vulnerability. Nevertheless, the deployment situation of these features in the webmail world is still unclear. To this end, we perform the first end-to-end and large-scale measurement of webmail service. For the server side, we first build an email address set with a size of 2.2 billion. Then we construct two webmail domain datasets: one contains 21k domains filtered from the email address set; the other only includes 34 domains but supports more than 75% of the 2.2 billion email addresses. After performing a comprehensive measurement on these two webmail domain datasets, we find that some features are poorly deployed. Furthermore, we also rank servers by analyzing the properties of HTTPS-related features. For the client side, we investigate implement of HTTPS-related features in 50 different combinations of web browsers and operating systems (OSes). We find that even the latest browsers have poor support for some features. For example, Firefox in all OSes does not support CT. Our findings highlight that the full deployment of the security features for the HTTPS ecosystem is still a challenge, even in the webmail service.

Index Terms—Email, Webmail Service, Certificates, HSTS, Expect-CT, DANE-TLSA, CT, SCSV, Revocation, Web browsers

1 Introduction

Despite the growth of mobile messengers and communication platforms, webmail is still an integral part of daily online life. However, many incidents indicate that the HTTPS protocol cannot alone preserve the confidentiality, integrity, and authenticity of messages over the Internet. One famous example is the compromise of DigiNotar [1], where the certificate authority issued 531 forged certificates that are the trusted root of the HTTPS protocol, hence giving chances to the man-in-the-middle attacks.

To alleviate the above situation, many security measures and extensions to reinforce the HTTPS ecosystem have been proposed. For instance, HTTP Strict Transport Security (`HSTS`) [2], HTTP Public Key Pinning (`HPKP`) [3], and `Expect-CT` [4] are added to the HTTP header. We term the three measures as `HTTP headers`, where HPKP has been deprecated due to complexity [5]. Furthermore, the DNS-based Authentication of Named Entities (DANE) was proposed to establish TLS connections via published TLS Authentication (`TLSA`) records [6]. This process does not rely on third-party CAs but on Domain Name System Security Extensions (`DNSSEC`) [7] to guarantee integrity and authority. We also have Certificate Transparency (`CT`) [8] and Signaling Cipher Suite Value (`SCSV`) [9] to mitigate the threat of CA compromise and TLS downgrade attacks. Additionally, certificate revocation mechanisms can invalidate misissued or malicious certificates before they expire, including `CRL` [10], `OCSP` [11], `OCSP Stapling` [12], and `OCSP Must-Staple` [13]. However, several previous studies [14]–[18] demonstrated that these HTTPS-related features were not deployed appropriately in general. The question then arises as to what the current situation for webmail services is.

In this paper, we present the first end-to-end and large-scale measurement of the HTTPS-related features for the webmail service, including webmail servers and web browsers. The contributions of this paper can be summarized as follows.

- We build an email address set that contains more than 2.2 billion unique entries from the Internet. To the best of our knowledge, it is the largest public email address set to date. Using this set, we comprehensively measure the deployment of HTTPS-related features on the filtered about 27 million domains, which contain more than 21k domains that provide the webmail service.
- We explore the joint use of the security features for the HTTPS protocol in 34 popular webmail servers, supporting more than 1.65 billion email addresses in our email address set. Furthermore, we propose a method for classifying server security levels according to the protection responsibilities and security benefits of the mechanism.

- *R. Li, Z. Zhang, and J. Shao are with the School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou, China 310018. E-mail: chn.junshao@gmail.com.*
- *R. Li, J. Shao, and X. Jia are with the key laboratory of network assessment technology, CAS (Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093).*
- *R. Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, Canada E3B5A3. E-mail: rlu1@unb.ca.*
- *X. Jia is the Institute of Information Engineering, Chinese Academy of Sciences, China. E-mail: jiaxiaoqi@iie.ac.cn*
- *G. Wei is with the School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou, China 310018. E-mail: weigy@mail.zjgsu.edu.cn.*
- *J. Shao is the corresponding author.*

- At last, we inspect in detail the implementation of HTTPS-related features in different versions of Chrome, Firefox, Edge, IE, Opera, QQ, 360, and Safari on Windows, Linux, macOS, Android, and IOS for a total of 50 combinations.

Overall, our findings show that HTTPS-related features of the webmail service are surprisingly poor. As a result, the notorious man-in-the-middle attacks could still work on the webmail service, and our main findings are as follows.

- We find that only 10.68% of the selected 21k webmail servers present their `HSTS` header, and only 32.14% of them support `OCSP Stapling`. Additionally, the deployment situation of the 27 million servers from our email address set is also serious. For instance, 48.59% of server certificates are invalid.
- For 34 popular webmail servers, we find that only 18 support TLSv1.3 and only nine deploy `OCSP Stapling`. Furthermore, we find that only 61.87% of the selected 21K webmail servers satisfy the minimum security level requirements.
- Firefox in all operating systems (OSes) does not support `CT`, and all browsers in Android and IOS don't send certificate status requests to `OCSP` servers. Unfortunately, some older browsers have extremely poor implementations of HTTPS-related features. For example, QQ 9.0 in Windows hardly detects certificate errors.

The rest of the paper is organized as follows. Section 2 provides the relevant technical background of the HTTPS ecosystem. After that, Section 3 introduces our dataset collection and server measurement process. Section 4 presents measurement results on the webmail server side. In what follows, we present the implementation of HTTPS-related features of web browsers in Section 5. Section 6 investigates related work. We then discuss the evolution of security feature deployment and future work in Section 7. At last, we give the conclusion in Section 9.

2 Background

In this section, we briefly describe the webmail delivery process and the security extensions of the HTTPS ecosystem we investigate in our paper.

2.1 Webmail Path

We give a typical email delivery process in Figure 1. More specifically, an email is firstly transmitted from the sender to his/her email service provider via the dedicated mail client or a web browser. After the sending processing inside the sender's provider, the email is transmitted to the receiver's provider, which will perform the receiving processing on the received email before the receiver's access. At last, the receiver can obtain the email via the dedicated mail client or a web browser. As we mentioned before, we focus on the configurations of the HTTPS protocol and its features, especially their use in the communication between the web browser and the webmail server.

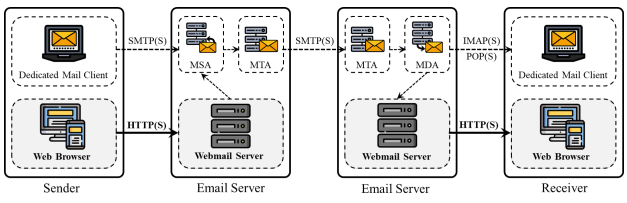


Fig. 1: Webmail path of an email message from sender to receiver.

2.2 HTTP Headers

Three HTTP header values named HTTP Strict Transport Security (`HSTS`) [2], HTTP Public Key Pinning (`HPKP`) [3], and `Expect-CT` [4] are proposed to mitigate various vulnerabilities of the HTTPS ecosystem. In particular, the first value enforces browsers to use HTTPS connections to resist the protocol downgrade attack and cookie hijacking. The second value pins the public keys to prevent man-in-the-middle attacks caused by fraudulent and mis-issued certificates. Unfortunately, HPKP has been deprecated due to its complexity and the serious harm caused by misconfiguration [5]. The last value allows the server to request the browser to ensure all of its certificates are properly logged by CT logs. It is worth mentioning that some browsers like Chrome and Safari already enforce the CT policy independently of Expect-CT [19], [20].

2.3 CT

Certificate Transparency (`CT`), described in an experimental RFC 6962 [8], is a framework to detect the compromise of Certificate Authority (CA). In this framework, users or CAs submit certificate chains to one or more public logs run by independent organizations, such as Google, DigiCert, and Symantec, who will return a corresponding Signed Certificate Timestamp (`SCT`). Since the `CT` log is append-only, tamper-resistant, and public, everyone can use the validity of the `SCT` to detect whether `CT` logs log the certificate. Thus, `CT` improves the chances of detecting problematic or fraudulent certificates. It is easy to see that we can check whether a server supports `CT` by asking for its `SCT`. Generally speaking, the server can deliver the `SCT` in HTTPS via the extension of the certificate, TLS, or Online Certificate Status Protocol (`OCSP`) `Stapling` [11].

2.4 DNSSEC

It is common sense that we cannot fully trust the data the original domain name system (DNS) provides due to the absence of security guarantees. To solve this problem, DNS Security Extensions (`DNSSEC`) were introduced in RFC 2065 [7], and the chain of trust model is the core component to check the data integrity and authentication in the DNS. In particular, the public key of a zone (recorded in the DNSKEY field) can be verified by using the Delegation Signer (`DS`) recorded in its parent zone, whose public key can be verified by the `DS` in the higher zone. According to the verification result of the signature on the data, we have the following four cases.

- *Secure*: The signature passes the verification, and the underlying public key can trace back to another public key the verifier trusts.

- *Insecure*: The signature passes the verification, and the chain from the underlying public key to the public key the verifier trusts exists. However, some DS is missing along the chain.
- *Bogus*: Though the chain from the underlying public key to the public key the verifier trusts exists, the signature cannot pass the verification.
- *Indeterminate*: There is no such chain from the underlying public key to the public key the verifier trusts. It is the default operation mode.

2.5 DANE-TLSA

Unreliable CAs have long been a pain point in the certificate ecosystem. To alleviate this situation, DNS-based Authentication of Named Entities (DANE) [6] was proposed to not rely on third-party CAs to verify server identities. The client first uses DNSSEC to confirm the authenticity of the TLS Authentication (TLSA) record published by the server. As mentioned before, if the DNSSEC verification result is "*Secure*", then the client can verify that the server certificate is consistent with the TLSA record according to the indication of the TLSA field.

2.6 Downgrade Protection

The Signaling Cipher Suite Value (SCSV) [9] was designed to avoid the security issues caused by the downgrade attacks in the context of TLSv1.2 and below. For instance, the adversary would seduce browsers to use down-level protocols, such as TLSv1.0. By using TLS_FALLBACK_SCSV in the ClientHello, the server will abort the connection if it can support a higher protocol than that the browser provides. TLSv1.3 incorporates its own downgrade protection mechanism by embedding particular values into ServerHello.random when negotiating TLSv1.2 or lower [21]. Meanwhile, the client should check for that particular value and abort the TLS handshake if it matches.

2.7 Certificate Revocation

As stated in X.509 PKI, the client should check the certificate's revocation status before using the corresponding public key. At an early age, the client usually appeals to the Certificate Revocation List (CRL) [10]. However, this approach gives a relatively heavy communication burden to the client. To solve the problem, the OCSP provides a revocation status query service for clients, while this method reveals the privacy of clients (such as browsing behavior). The OCSP Stapling [12] is further proposed to reduce the probability of privacy leaks. In particular, the server retrieves the revocation status from the OCSP responder and forwards it to the client directly. Unfortunately, the client may choose to continue a connection in this case even if the server does not provide the OCSP response. To address this problem, another extension named OCSP Must-Staple [13] is proposed. It asks the client to abort the connection if it does not receive a valid OCSP response.

3 Data Collection

In this section, we introduce the dataset collection process and server-side measurement methods. Our study aims to measure the configurations of the HTTPS protocol and its security features in webmail services. The first problem we should solve in our measurement is the lack of a large-scale public webmail domain dataset. To this end, we extract more than 2.2 billion unique email addresses from the Internet. Afterward, we carefully build a list of webmail domains from the email address dataset and measure their HTTPS-related features.

TABLE 1: The leaked databases used in this paper.

Dataset	Year	Storage Size	Email Addrs*
Adobe	2013	9.3GB	152M
Exploit.In	2016	25GB	593M
Anti Public	2016	120GB	457M
Collection#1	2019	90GB	772M
Facebook	2019	90.2GB	509M
Wattpad	2020	120GB	268M
Total	-	454.5GB	2700G
Filter Results	-	48GB	2200G

* Email Addrs indicates the number of email addresses.

3.1 Datasets

As we mentioned before, the dataset is the first obstacle we should go over in this study. The ideal one should contain webmail domains that can be broadly representative of global webmail services. One crucial criterion of representativeness is how many email accounts are in the collected webmail domains. Based on this observation, we collect as many email addresses as possible. In particular, we compile an email address set from six publicly leaked datasets as shown in Table 1, including Adobe leak and five sets suggested by the largest breaches list on https://haveibeenpwned.com/ [22]. From these datasets with 454.5GB storage, we get raw data of email addresses with 169GB storage. After deduplication, we obtain 2.2 billion unique email addresses with 48GB storage. By using regular expression matching, we compose 27 million domains, which comprise the dataset we call Dataset-domain.

At first glance, Dataset-domain is the webmail domain set we want. However, it is unfortunately incorrect. For instance, from the email address bob@yahoo.com, we can get the domain yahoo.com, which is not yet a webmail domain. Therefore, Dataset-domain is just a collection of email server domains.

Generally speaking, a webmail domain name is in the form of mail.* ... (e.g., mail.yahoo.com). Therefore, to filter out webmail domains from Dataset-domain, we first choose domain names containing "mail" as a candidate domain list. Then we apply keyword matching on web content to build a new candidate domain list[1]. Specifically, the web content of the candidate domain list is obtained by crawling web pages with go-colly [23]. As for the keyword list, we build it through the following steps. We first construct 38 keywords by observing some webmail pages. After that, we remove inappropriate keywords by testing different keyword combinations. That is, if adding a keyword causes many false positives or false negatives in webmail domain matching results, we remove it. After repeated tests, we carefully select 16 webpage text keywords (such

1. This process automatically handles language and capitalization

as "mail" and "mailbox") and four webpage element keywords (such as "login" and "send"). Additionally, with further extensive experiments, we can identify most webmail domains when the web content contains 10 out of 20 keywords. Ultimately, we obtain a new candidate domain list containing 24K domains.

To ensure the accuracy of the webmail domain dataset, we then validate the new candidate domain list using FortiGuard [24] and NetSTAR [25] web filtering tools. Specifically, we reserve domains whose category is Web-based Email (88%). For the remaining domains (12%), we manually visit the webpage to check whether it is a webmail domain. After the above steps, we assemble a webmail domain dataset with a size of 21k, which we call Dataset-webmail.

At last, to better understand the security status of webmail services for most users, we also construct a dataset of popular webmail domains. The detailed process is as follows. Firstly, when resolving email addresses from 2.2 billion addresses, we rank the domains according to the frequency of appearance and call the resulting ranking as *domain-ranking*. Then, we select the top 54 domains from the *domain-ranking*, and manually identify the relevant webmail domains. However, we find that several domains use the same webmail service in the same webmail domain. For instance, all of yahoo.co.uk, yahoo.fr, and ymail.com use mail.yahoo.com as their webmail service domain. Therefore, we merge the different domains using the same webmail service and remove those that do not provide the webmail service. In addition, to construct as complete a dataset of popular webmail domains as possible, we also add seven popular webmail domains: mail.protonmail.com, mail.zoho.in, fastmail.com, runbox.com, www.icloud.com, mail.smt.docomo.ne.jp, and mail.sina.com.cn, according to the data in [26] and the statistics in [27]. In the end, a set of 34 popular webmail domains, supporting 1.65 billion email addresses, is obtained. We call this small dataset Dataset-34.

3.2 Measurement process

We conduct our active measurements to understand the practical deployment of various security features of the HTTPS ecosystem in webmail servers. As suggested by VanderSloot et al. [28], both IP address and domain name should be used in the active scans to reduce bias due to the case that many services run with the same IP address. In our TLS scans, the domain name is embedded in an extension of the ClientHello, called the Server Name Indication extension [29].

3.2.1 Scan List

There are only domain names in Dataset-domain, Dataset-webmail, and Dataset-34; hence we need to perform A record lookups for the underlying domain names by using an unmodified version of massdns [30]. After that, we still need to check the liveness of the domains. To this end, we make use of ZMap [31] to conduct tcp443 SYN-ACKs, and Internet Control Message Protocol (ICMP) scans to get a live IP address set as complete as possible. We union these two scanning results and create a list of (IP address, domain name) pairs called List-scan. Accordingly, we have three List-scans, named List-domain, List-webmail, and List-34, for Dataset-domain, Dataset-webmail, and Dataset-34, respectively. In the following, all server-side measurements are based on these three List-scans. Specifically, TLSA records are collected from DNS, and other data are mainly obtained via performing TLS negotiations with servers on port 443. Furthermore, we validate the server's certificate chain based on the Mozilla Root CA certificate [32] in the TLS handshake. The detailed measurement process of HTTPS-related features is as follows.

3.2.2 HTTP Headers

To analyze the deployment of the three HTTP headers mentioned in Section 2.2, we establish TLS connections according to the List-scan and intercept the corresponding headers using a modified version of Goscanner [33] (used in [34]). In particular, the fields Strict-Transport-Security, Public-Key-Pins or Public-Key-Pins-Report-Only, and Expect-CT are for HSTS, HPKP, and Expect-CT.

3.2.3 DANE-TLSA

We develop a domain name resolver based on the software named UNBOUND [35] to resolve TLSA records in List-scan and obtain DNSSEC verification results. After that, we reconnect to the server and fetch the certificate chain presented by the server if we can resolve a TLSA record. Finally, we complete the verification of DANE according to the Certificate Usage, Selector, Matching Type, and Certificate Association Data in the TLSA record and the certificate chain of the webmail server.

3.2.4 CT

We build a tool to collect the CT status of servers in List-scan. Specifically, the tool first fetches the SCT in certificate extensions, TLS extensions, and OCSP Stapling. Then it checks the validity of obtained SCTs based on the Google Chrome log list [36] by using the methods defined in RFC 6962. At last, the tool also determines the level of the certificates according to their Object Identifiers (OIDs).

3.2.5 Downgrade Protection

It is easy to check downgrade protection mechanisms. Particularly, we initiate TLS connections according to List-scan. Once the connection succeeds, we first check the server's TLS version. If the server supports TLSv1.2 or below, we immediately conduct a reconnection with the TLS_FALLBACK_SCSV cipher suite and downgrade the TLS version. If the server supports SCSV, the connection will be aborted. If the server supports TLSv1.3, we conduct a reconnection with the lower TLS version. After that, we will check whether the last 8 bytes of ServerHello.random is equal to the value specified in the RFC 8446.

3.2.6 Certificate Revocation

According to the specification in certificate revocation mechanisms, only OCSP Stapling is transported by the TLS extension. The others, including CRL, OCSP, and OCSP Must-Staple, are embedded in the certificate extensions (Issuing Certificate URL, CRL Distribution Points, and the extension the OID specifies, respectively). Hence, we can verify the status of certificate revocation according to the existence of the corresponding data.

3.3 Ethical Considerations

Due to the limitations of the experimental environment, only active scans are included in our daily scanning work—no need to consider the problem of leaking the private information of each part when passively collecting data. For our active scans, we strictly follow the scanning precautions outlined in [31]. We follow the best scanning practices, such as adjusting the query rate (rate-limit) according to the current network environment, which will end issues such as exhausting network resources and affecting upstream suppliers. As for data sharing issues, we will not release any datasets that contain personal privacy data to protect the privacy of involved parties.

4 Webmail Server Configuration

In this section, we first present the measurement results of HTTPS-related features on List-webmail. For better illustration, we also compare it with List-domain. Then we conduct a detailed analysis of the top-34 webmail server alone. At last, we also grade the security status of webmail servers according to HTTPS-related features.

4.1 List-webmail and List-domain

As we can see from Table 2, almost all the scanned webmail servers support TLS connections. However, it is dispirited that only 29.77% of the servers in List-webmail are using the newest version of the TLS protocol (TLS 1.3). Even 0.80% of webmail servers (339 domains) are using the deprecated version of TLS protocol (TLS 1.0) that contains many vulnerabilities, such as BEAST and Downgrade attacks [37].

Then, we analyze the validity of the certificate. As shown in Figure 2, we find that 40.13% of the servers in List-webmail could not pass the certificate verification. Invalid certificates are mainly expired and signed by an unknown CA. As the basic requirement for users to use the webmail service in the browser, the certificate configuration of the webmail server is far from our expectations.

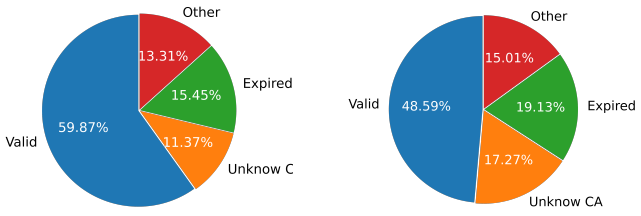
Regarding the six HTTPS-related security features listed in Section 2, we find that the deployment status of `HTTP headers` and `DANE-TLSA` is quite poor, while the situation of `CT`, `SCSV`, and certificate revocation is much better. We give the details below.

4.1.1 HTTP Headers

Table 3 gives the summary of our scanning results on `HTTP headers`. We can see that less than 13% of servers support at least one of the three `HTTP headers`. During our scanning process based on List-domain, we find that many servers supporting `HSTS` return incorrect headers. The typical errors include the extra right single quotation mark (16.93%) and the word `always` (13.90%). We also find that only three webmail domains, including `mail-4.de`, `shitmail.de`, and `mail.protonmail.com`, support `HPKP`. The potential reason for this situation is that Google deprecated it in 2018 due to its complexity [5]. At last, although there are only 1.12% of `Expect-CT` servers in List-webmail setting invalid `max-age` value, it is the worst one among the three `HTTP headers`. We also note a webmail domain (`mail.protonmail.com`), an encrypted webmail service [38], setting all three `HTTP headers`.

To resist `HTTP headers` related attacks, such as SSL stripping attacks [39], three attributes, named `max-age`, `includeSubDomains`, and `Pre-load` are added to the HTTP header. We can find the support ratios of these three attributes in Table 3, and they are not as good as expected.

Max-age. The `max-age` attribute denotes how long the browser can directly use the cached `HTTP headers`. Table 4 shows a non-negligible probability of receiving invalid `max-age` values, such as NaN and non-numerical values. Figure 3 shows the distribution of valid `max-age` values for List-webmail and List-domain, respectively. More than 60% of valid `max-age` values for `HSTS` in the webmail servers are set as 31536000 (365 days) and 15768000 (182 days) as specified in RFC 6797 [2]. In contrast, the ratio corresponding to List-domain drops to 40%, and only a small part of valid values for `Expect-CT` are the ones in the related draft RFC [4], no matter for List-webmail or List-domain. Among the valid values, we find that several popular servers set `max-age=0`, like `mail.daum.net` [40], one of the largest portal sites and opening the first email service in South Korea. As specified in the related draft RFCs [2]–[4], when `max-age` is set as zero, the browser must delete the policy related to the underlying `HTTP headers`, causing vulnerabilities due to



(a) List-webmail (b) List-domain

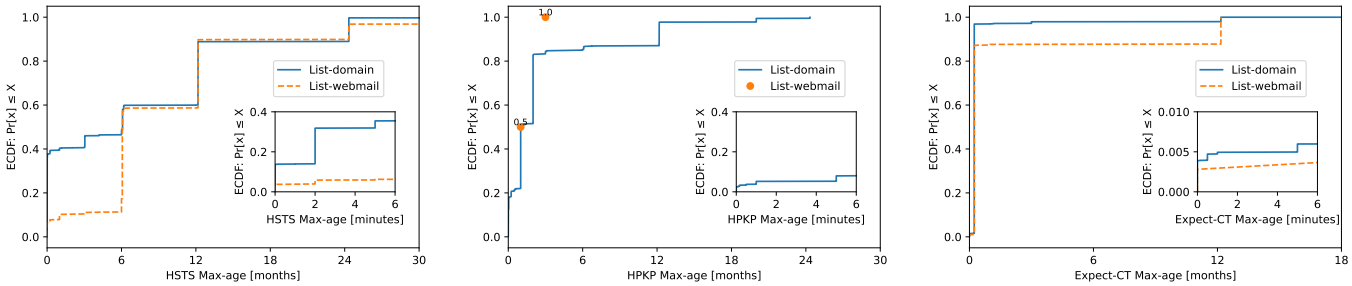
Fig. 2: Certificate validity statistics from our active scans.

TABLE 2: Statistics of different TLS versions seen in our active scans.

TLS Version	List-webmail	List-domain
TLSv1.3	29.77%	55.75%
TLSv1.2	69.43%	43.41%
TLSv1.1	-	-
TLSv1.0	0.80%	0.84%

TABLE 3: Statistics of three `HTTP headers`.

	List-webmail	List-domain
`HSTS`	10.68%	9.28%
`max-age`	10.68%	9.28%
`includeSubDomains`	6.36%	2.36%
`Pre-load`	1.83%	1.13%
`Pre-load and embedded`	0.10%	0.05%
`HPKP`	0.01%	0.01%
`max-age`	0.01%	<0.01% (1465)
`includeSubDomains`	<0.01% (1)	<0.01% (1350)
`Expect-CT`	3.26%	4.19%
`max-age`	3.24%	4.18%
`Pre-load`	0.02%	<0.01% (61)
`Pre-load and embedded`	-	<0.01% (1)
`enforce`	0.04%	0.07%



(a) HSTS (b) HPKP (c) Expect-CT

Fig. 3: Distribution of `max-age` attribute for three `HTTP headers`.

TABLE 4: Invalid values in the `max-age` attribute of three `HTTP headers`.

	List-webmail			List-domain		
Headers	HSTS	HPKP	Expect-CT	HSTS	HPKP	Expect-CT
NaN	-	-	0.56%	0.01%	-	0.02%
Non-numerical	-	-	0.56%	0.06%	21.91%	0.04%
Total	-	-	1.12%	0.07%	21.91%	0.06%

SSL stripping attacks [39].

includeSubDomains. `includeSubDomains` is an optional attribute that enables subdomains to also be protected by `HTTP headers` [2]. It only supports `HSTS` and `HPKP`, but not `Expect-CT`. From Table 3, we can see that only a tiny portion of (webmail) servers support `includeSubDomains` no matter for List-webmail or List-domain, especially, only 1 and 1350, respectively.

Pre-load. The `Pre-load` attribute denotes that the related policy should already be embedded in the browser, which allows users to be protected by `HTTP headers` the first time they visit the domain [2]. It could work only if the browser contains the related information indeed. Hence, we need to cross-check the configuration of the underlying browser and the HTTP header the server sends. To this end, we use Chrome's preload list [41], which many mainstream browsers integrate into their preload lists, such as Firefox [42] and Edge [43]. By using this list, we investigate the pre-loading condition of `HSTS` and `Expect-CT`. As we can see from Table 3, only 22 (0.10%) webmail servers in List-webmail and 9k (0.05%) servers in List-domain are included in the `Pre-load` list and set the `Pre-load` directive for `HSTS` at the same time. The numbers for `Expect-CT` are even lower, 0 and 1 for List-webmail and List-domain, respectively. The major browsers use `HPKP Pre-load` lists, such as Mozilla's `HPKP` pre-loading list [44], which includes some high profile sites (e.g., Google, Facebook, Twitter).

Public-Key-Pins. It is worth mentioning that Public-Key-Pin is the core component of `HPKP`. The browser can use it to verify the server certificate's validity, preventing attackers from using improper or fraudulent certificates to carry out man-in-the-middle attacks. However, we find that most of the servers supporting `HPKP` (86.09%) cannot successfully pass the verification of pins. If the `max-age` value of HPKP is set too large, the browser may deny normal access to the domain for a long time. This situation may explain the fact that `HPKP` is deprecated now [5].

4.1.2 DANE-TLSA

After obtaining the data related to `TLSA`, we also need to perform `DNSSEC` verification to verify their validity. Specifically, we use `UNBOUND` to resolve the related records, such as `DNSKEY` containing the `DNSSEC` public keys, `RRSIG` containing the cryptographic signature on `RRsets` signed by the superior's private key, and the hash values `DS` of `DNSKEY`s. Recall the `DNSSEC` part in Section 2.4. We have that `DNSSEC` verification passes if we get the "*Secure*" verification result. At last, we check whether `TLSA` records are consistent with certificates.

According to our measurements, the support ratio of `DANE-TLSA` is surprisingly low. In particular, as described in Table 5, only 0.16% (resp. 0.08%) of servers in List-webmail (resp. List-domain) support `TLSA`, only 85.29% (resp. 72.53%) of these servers can return the "*Secure*" verification result, and only 67.65% (resp. 62.17%) of these servers can pass `DANE` verification.

We find that 18 webmail servers cannot pass `DANE` verification due to the following two cases. 1) We get "*Insecure*" `DNSSEC` verification results for nine servers. Especially, the signed `TLSA` records are absent in five out of the nine servers, and the `DS` records are missing in all of them. 2) We get 12 webmail servers failing to provide certificates matching `Certificate Association Data Field` in the `TLSA` records. The potential reason for this situation is that the operator manually updates the certificate but does not promptly update the associated data in the `TLSA` records.

TABLE 5: The ratio of `TLSA` record, and the results of `DNSSEC` and `DANE` validation.

	List-webmail	List-domain
TLSA	0.16%	0.08%
DNSSEC	0.16%	0.08%
Secure	0.13%	0.06%
Insecure	0.02%	0.02%
Bogus	-	<0.01% (10)
Indeterminate	-	-
with signed	0.14%	0.06%
without signed	0.01%	0.02%
with DS	0.13%	0.06%
without DS	0.02%	0.02%
DANE valid	0.10%	0.05%

TABLE 6: CT detailed statistics from our active scans.

	List-webmail	List-domain
Domain w/SCT	93.68%	93.54%
valid	90.02%	90.50%
via X.509	90.02%	90.50%
via TLS	-	<0.01% (22)
via OCSP	0.02%	<0.01% (424)
Each SCT (valid)	91.01%	91.60%
via X.509	90.98%	91.60%
via TLS	-	<0.01% (12)
via OCSP	0.03%	0.01%

From the collected metadata, we find some interesting cases for List-webmail (resp. List-domain). For example, 83.33% (resp. 87.14%) of the Certificate Usage Field in the TLSA records point to domain-issued certificates, which is complied with that in [14]. According to Selector Field in TLSA records, we also find that 35.90% and 64.10% of webmail servers (60.62% and 39.02% of domains) supporting TLSA store the full certificate and only the public key (SubjectPublicKeyInfo), respectively.

4.1.3 CT

As mentioned in Section 2.3, SCT indicates that the certificate has been logged in CT logs, and it is usually embedded in the certificate extension, TLS extension, or OCSP Stapling. Table 6 shows that most servers support CT. It is not surprising that CT gets the best configuration among all the security features we investigate in this study, since Chrome would not trust the certificate issued after April 2018 but failing to pass the CT verification [19]. Furthermore, we find that most (webmail) servers embed SCTs in the certificate extension, only four webmail servers and 424 servers embed them in the OCSP Stapling, and no webmail server and 22 servers embed it in the TLS extension. The main reason for this delivery situation is that delivering the SCT via a certificate only requires the effort of the CA, while the other two ways shift the burden to the server operator.

As we can see from Table 6, 8.99% (resp. 9.40%) of SCTs we collected for List-webmail (resp. List-domain) are invalid, and most of them are sent via X.509 certificates. This motivates us to conduct an in-depth analysis of invalid SCTs with valid certificates. We find that 122 webmail servers comply with this feature, and all of these SCTs are embedded in X.509 certificates. Most of the 122 certificates are issued by Let's Encrypt (52), Sectigo Limited (26), and TAIWAN-CA (13).

We also investigate the status of CT logs and their operators. As we can see from Table 7, most certificates are logged in Google 'Argon2021' no matter for List-webmail or List-domain, and the CT logs Google operates contains 45.62% and 46.82% of servers for List-webmail and List-domain, respectively. It is not surprising for this situation since Google is the one proposing the concept of CT. Moreover, as stated in [45], for certificates issued on-or-after April 15th, 2022, Google delegates the power of logging certificates to other CT logs. Considering EV certificates. On 1st Jan. 2015, Google stipulated that EV certificates must be submitted to at least one SCT from Google-operated log and one from non-Google-operated log [46]. Thus, we explore the diversity of

TABLE 7: Different logs and log operators for SCTs.

	List-webmail	List-domain
Logs Description		
Google 'Argon2021'	21.16%	16.35%
Cloudflare 'Nimbus2021'	16.28%	10.25%
Google 'Xenon2021'	10.78%	17.12%
DigiCert 'Yeti2021'	8.14%	9.85%
Google 'Argon2022'	7.12%	6.97%
DigiCert 'Yeti2022'	7.03%	7.32%
Let's Encrypt 'Oak2021'	5.56%	8.62%
Google 'Xenon2022'	3.48%	4.10%
DigiCert 'Nessie2022'	3.47%	3.06%
Let's Encrypt 'Oak2022'	2.81%	3.17%
Cloudflare 'Nimbus2022'	1.97%	2.10%
Sectigo 'Sabre' CT	1.83%	1.34%
Google 'Rocketeer'	1.75%	0.76%
Sectigo 'Mammoth' CT	1.64%	2.23%
Total	93.02%	93.24%
Log Operators		
more than one	93.34%	92.94%
only one	2.66%	3.81%
none	3.90%	3.25%
Logs Diversity		
more than one	93.34%	92.94%
only one	2.66%	3.81%
none	3.90%	3.25%
EV		
more than one	1.84%	2.93%
only one	0.01%	0.02%
none	0.04%	0.02%
google and non-google	1.83%	2.93%

logs and find that more than 90% of the certificates (even more than 95% of the EV certificate) satisfy the multiple logs requirement.

There are generally three assurance levels for certificates issued by mainstream CAs: Domain Validation (DV), Organization Validation (OV), and Extended Validation (EV) [47]. EV presents the highest standard of trust and requires more verification and review, such as requesting the entity's legal identity before certificate issuance. When measuring the deployment of CT, we also investigate the assurance levels of the certificates based on the unique Object Identifier. Table 8 shows that most of the certificates are the lowest assurance level DV, and only a small portion of them are EV, even for the webmail servers in List-webmail. The poor support of EV certificates may be because web browsers no longer visually display the higher security flag (green address bar) for websites equipped with the EV certificate [48], [49].

4.1.4 Downgrade Protection

To investigate SCSV in TLSv1.2 and below, we downgrade the protocol after the first handshake and carry it to initiate a second connection on purpose, retaining the server's response and errors. We classify the received connection errors into three types: (a) The server deploys SCSV, terminates the connection normally, and returns remote error: tls:.

TABLE 8: Certificate level statistics from our active scans.

	List-webmail	List-domain
DV	58.79%	77.25%
OV	33.61%	14.25%
EV	1.76%	2.78%
Unknown	5.84%	5.72%

TABLE 9: SCSV statistics from active scans.

	List-webmail	List-domain
Abort	92.33%	92.08%
Continue	3.83%	2.98%
Other	3.84%	4.94%

(b) The server continues the TLS handshake and uses the downgraded version. (c) The connection is unsuccessful due to other reasons, such as i/o timeout and EOF.

According to the above classification, we get the results in Table 9. It is depressing that there are still 3.83% (2.98%) of servers in List-webmail (resp. List-domain) does not support SCSV. We extract the TLDs of all the webmail servers that fail to support SCSV. We find that most TLDs are in the .com zone (299), .net zone (51), .edu zone (44), and .org zone (33). We hope these zones can attract attention and improve their deployment. For TLSv1.3 downgrade protection, we find 90.19% (resp. 87.15%) of servers in List-webmail (resp. List-domain) correctly implement RFC 8446.

Finally, we surprisingly find that 459G (17.93%) email addresses are handled by web servers vulnerable to downgrade attacks.

4.1.5 Certification Revocation

Table 10 shows the deployment status of certificate revocation of the servers in List-webmail and List-domain. We find that most of the servers support at least one certificate revocation method. The OCSP method is the most popular, while the OCSP Must-Staple gets the least support due to the high deployment complexity. In particular, only adminmail.ru, dublinmaildrop.com, mail.upla.edu.pe, maileverything.com, and mail.opera.com support the OCSP Must-Staple. Furthermore, the server in List-webmail has significantly lower support for OCSP Stapling compared to List-domain. This can potentially affect the performance of the webmail service. Finally, among webmail servers that support certificate revocation, we find that 14% of them could not obtain the certificate's revocation status through the supported revocation mechanism. This suggests that the availability of certificate revocation needs to be further improved.

4.2 List-34

We also investigate the joint use of HTTPS security features in popular webmail servers according to List-34. As shown in Table 11, we find that almost all the popular webmail servers support CT, SCSV, and certificate revocation, while HTTP headers and DANE-TLSA get rare attentions. In particular, none of them support TLSA, and only mail.yahoo.com, mail.aol.com, mail.protonmail.com, and www.icloud.com support Expect-CT. For the assurance levels of certificates, we can see that only four webmail servers use the highest level EV, and most of them support DV and OV. On the bright side, all the webmail servers use valid certificates, and the default TLS version is higher than 1.1. However, one startling finding is that outlook.live.com does not support SCSV, hence suffering from downgrade attacks. Considering the certificate revocation mechanism, we find that 25 webmail servers do not support OCSP Stapling, which will undoubtedly increase the latency of accessing these webmail servers. Additionally, no webmail servers support OCSP Must-Staple, which may be related to client soft-failure and unreliable revocation responses [50].

TABLE 10: Statistics of four certificate revocation methods from active scans.

method	List-webmail	List-domain
CRL	48.90% (52.28%)*	36.16% (37.46%)*
OCSP	94.68% (99.95%)*	94.52% (99.94%)*
OCSP Stapling	16.92% (18.86%)*	32.14% (35.45%)*
OCSP Must-Staple	0.02% (0.03%)*	0.03% (0.03%)*

* The ratios in parentheses are on the premise that the certificate is valid.

According to the above findings, the active scan result based on List-webmail is corroborated by that based on List-34. The webmail servers, even the popular webmail servers, should pay more attention to the deployment of HTTPS security features.

4.3 Security Levels Analysis

According to [48], each HTTPS-related feature has its specific protection responsibilities, and only mutual cooperation can ensure the security of the HTTPS ecosystem. As shown in Table 12, we classify four security levels according to the protection responsibilities and security benefits of the feature, which more intuitively depicts the security state of the webmail world. In the following, we will detail our classification method.

Considering the protection responsibility. We classify certificate, HPKP, and DANE-TLSA as Authentication Credentials; CT and Expect-CT as Mis-Issuance protection; TLS downgrade protection, and HSTS as downgrade protection. Furthermore, we consider a webmail server to support Certificate Revocation if it deploys at least one certificate revocation mechanism.

Considering the security benefit, which is the comprehensive consideration of security protection and deployment complexity. *First*, TLS versions less than 1.2 have been deprecated due to their vulnerabilities, so we consider they have no security benefits. *Second*, certificates are the most common authentication credential but depend on the CA for reliability. DANE-TLSA can solve this problem, but its complexity causes the server difficult to deploy correctly. Therefore, we consider the security benefit of the certificate is higher. Furthermore, deprecated HPKP may lockout customers for long periods when used improperly, so we consider it has no security benefits. *Third*, CT is the most popular framework for Mis-Issuance protection and generally does not burden server operators, while Expect-CT ensures the implementation of the CT framework. Also, browsers that enforce CT don't care if the server supports Expect-CT. Therefore, we believe the security benefit of CT is higher. *Fourth*, the server operator requires more effort when configuring HSTS compared to TLS downgrade protection that can be automatically implemented by software like OpenSSL. Therefore, we consider the security benefits of TLS downgrade protection are slightly higher.

Based on the above analysis, we consider that the webmail server should implement as many different protection

TABLE 11: HTTPS-related configurations of 34 popular webmail servers.

Webmail Server	PCT[1]	PCTSUM[1]	Mail Provider	Protocol	CertLevel[2]	CertValid[2]	HTTP Headers			DANE-TLSA			CT[4]		TLSDownPro[5]	CertRev[6]
							HSTS	Expect-CT	DepHPKP[3]	w/TLSA	DNSSEC	DANE	SCT	Valid		
mail.yahoo.com	15.97%	15.97%	yahoo.com	TLSv1.3	OV	✔	✔	✔	✔	✗	✗	✗	★	✔	✔	▲■♦
mail.google.com	14.59%	30.56%	gmail.com	TLSv1.3	DV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
outlook.live.com	13.9%	44.46%	hotmail.com	TLSv1.2	OV	✔	✔	✗	✔	✗	✗	✗	★	✔	✗	▲■♦
e.mail.ru	11.87%	56.33%	mail.ru	TLSv1.2	OV	✔	✔	✗	✔	✗	✗	✗	★	✔	✔	▲■
mail.rambler.ru	5.56%	61.89%	rambler.ru	TLSv1.2	OV	✔	✔	✗	✔	✗	✗	✗	★	✔	✔	▲■♦
mail.yandex.ru	4.04%	65.93%	yandex.ru	TLSv1.3	OV	✔	✔	✗	✔	✗	✗	✗	★	✔	✔	▲■
mail.aol.com	2.84%	68.77%	aol.com	TLSv1.3	OV	✔	✔	✔	✔	✗	✗	✗	★	✔	✔	▲■♦
web.de	0.70%	69.47%	web.de	TLSv1.3	OV	✔	✔	✗	✔	✗	✗	✗	★	✔	✔	▲■
mail.qq.com	0.66%	70.13%	qq.com	TLSv1.2	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
xfinity.com	0.59%	70.72%	comcast.net	TLSv1.2	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✗	▲■
gmx.net	0.59%	71.31%	gmx.de	TLSv1.3	OV	✔	✔	✗	✔	✗	✗	✗	★	✔	✔	▲■
mail.yahoo.co.jp	0.40%	71.71%	yahoo.co.jp	TLSv1.3	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
mail.163.com	0.35%	72.06%	163.com	TLSv1.3	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
mail.libero.it	0.29%	72.35%	libero.it	TLSv1.2	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
email.seznam.cz	0.27%	72.62%	seznam.cz	TLSv1.3	DV	✔	✔	✗	✔	✗	✗	✗	★	✔	✔	■
mail.lycos.com	0.26%	72.88%	lycos.de	TLSv1.2	DV	✔	✔	✗	✔	✗	✗	✗	★	✔	✔	■
mail.epost.de	0.26%	73.14%	epost.de	TLSv1.2	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✗	▲■
email.wp.pl	0.23%	73.37%	wp.pl	TLSv1.3	DV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
myspace.com	0.23%	73.60%	Myspace	TLSv1.2	DV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
mail.naver.com	0.18%	73.78%	naver.com	TLSv1.2	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■♦
mail.com	0.17%	73.95%	mail.com	TLSv1.3	OV	✔	✔	✗	✔	✗	✗	✗	★	✔	✔	▲■
poczta.interia.pl	0.17%	74.12%	interia.pl	TLSv1.2	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
orange.fr	0.16%	74.28%	orange.fr	TLSv1.3	OV	✔	✔	✗	✔	✗	✗	✗	★	✔	✔	▲■
webmail.cox.net	0.16%	74.44%	cox.net	TLSv1.2	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
verizon.com	0.16%	74.60%	verizon.net	TLSv1.2	EV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■♦
mail.126.com	0.15%	74.75%	126.com	TLSv1.3	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
free.fr	0.15%	74.90%	free.fr	TLSv1.2	DV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
mail.protonmail.com	-	-	protonmail.com	TLSv1.3	EV	✔	✔	✔	✔	✗	✗	✗	★♠	✔	✔	▲■♦
mail.zoho.com	-	-	zoho.in	TLSv1.3	DV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■
fastmail.com	-	-	fastmail.com	TLSv1.3	OV	✔	✔	✗	✔	✗	✗	✗	★	✔	✔	▲■♦
runbox.com	-	-	runbox.com	TLSv1.2	EV	✔	✔	✗	✔	✗	✗	✗	★	✔	✗	▲■
www.icloud.com	-	-	www.icloud.com	TLSv1.3	EV	✔	✔	✔	✔	✗	✗	✗	★	✔	✔	▲■♦
mail.smt.docomo.ne.jp	-	-	smt.docomo.ne.jp	TLSv1.2	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✗	■
mail.sina.com.cn	-	-	sina.com.cn	TLSv1.3	OV	✔	✗	✗	✔	✗	✗	✗	★	✔	✔	▲■

[1] PCT and PCTSUM indicate the frequency and cumulative of mail providers.
[2] CertLevel and CertValid indicate the certificate level and certificate validity.
[3] DepHPKP indicates the deprecated HPKP.
[4] Three ways of transmitting SCTs: ▶ indicates TLS extensions, ★ indicates X.509 extensions, and ♠ indicates OCSP Stapling.
[5] TLSDownPro indicates the TLS Downgrade Protection.
[6] CertRev indicates the four mechanisms for checking a certificate's revocation status: ▲ indicates the CRL, ■ indicates the OCSP, ♦ indicates the OCSP Stapling, and ▼ indicates the OCSP Must-Staple.

TABLE 12: Rating standards for the security levels of servers. ✔ denotes support, ✗ denotes no support. ○ denotes not considering. Except for the Level-C, the conditions of other levles must meet simultaneously.

Rating	TLSVer	AuthCred[1]			MisIssPro[2]		CerRev	DownPro[3]	
		CertValid	DepHPKP	DANE-TLSA	CT	Expect-CT		TLSDownPro	HSTS
Level-S	1.3	✔	✔	✔	✔	✔	✔	✔	✔
Level-A	1.3	✔	✔	○	✔	○	✔	✔	○
Level-B	1.2/1.3	✔	✔	○	○	○	○	○	○
Level-C	<1.2	✗	✗	○	○	○	○	○	○

[1] AuthCred indicates the Authentication Credential.
[2] MisIssPro indicates the Mis-Issuance Protection.
[3] DownPro indicates the Downgrade Protection.

responsibilities as possible. Furthermore, among mechanisms with the same protection responsibilities, the webmail server should support at least the one with higher security benefits. Therefore, we consider Level-B the minimum requirement for the webmail server to satisfy secure transmission. Specifically, the server should deploy the TLS version greater than 1.1, provide a valid certificate and not support HPKP. If either requirement is violated, we rank the server as Level-C, which cannot be trusted by Internet users to transmit email. If the webmail server deploys TLSv1.3, a valid certificate, CT, at least one certificate revocation mechanism, TLS downgrade protection, and does not support HPKP, we rank it as Level-A. If the webmail server further supports DANE-TLSA, Expect-CT, and HSTS, we rank it as Level-S.

TABLE 13: Statistics of servers in different security levels.

Rating	List-webmail	List-domain
Level-S	0 (-)	41 (<0.01%)
Level-A	5437 (24.80%)	3251753 (26.48%)
Level-B	8127 (37.07%)	2566529 (20.90%)
Level-C	8361 (38.13%)	6462313 (52.62%)

Table 13 shows the situation for different security levels of servers. Unfortunately, we find that Level-C accounted for the largest percentage in both List-webmail and List-domain, which should be alarming for server operators. Reassuringly, the proportion of Level-B servers in List-webmail is significantly higher than that of List-domain. Furthermore, about a quarter of webmail servers can meet the requirements of Level-A, but none can further satisfy the criteria of Level-S, which may be a decision made by server operators weighing the security benefits of the mechanism. In conclusion, we hope that webmail server operators can re-check the security configuration to at least satisfy the requirements of Level-B.

4.4 Summary

Considering the importance and confidentiality of webmail services, users sensitive to privacy and security prefer to choose webmail servers with a better HTTPS configuration. Therefore, improving the server security configuration can simultaneously enhance the prestige of the webmail domain and promote the development of HTTPS-related mechanisms. Our analysis can serve as an essential guide for investigating webmail server security.

TABLE 14: Web browser implementation of HTTPS-related features. ✔ denotes correctly identified certificate error; ✗ denotes incorrect. ● denotes the feature is supported; ○ denotes it is not supported.

OS	Browser	Version	TLSv1.3	DeTLSVer[1]	SelfSignCert[2]	ExpCert[2]	NoSANCert[3]	DomErrCert[3]	CRL/OCSP	OCSPStal[4]	OCSPMS[4]	CT	HSTS	HPKP	DANE-TLSA
Win.	Chrome	103	●	○	✔	✔	✔	✔	●	●	○	●	●	○	○
		80	●	○	✔	✔	✔	✔	○	○	○	●	●	○	○
		67	○	○	✔	✔	✔	✔	○	○	○	○	●	○	○
	Firefox	102	●	○	✔	✔	✔	✔	●	●	●	○	●	○	○
		80	●	○	✔	✔	✗	✔	●	●	●	○	●	○	○
		63	●	○	✔	✔	✗	✔	●	●	●	○	●	○	○
	Edge	103	●	○	✔	✔	✔	✔	●	●	○	●	●	○	○
		83	●	○	✔	✔	✔	✔	●	●	○	○	●	○	○
	IE	11	○	●	✔	✔	✗	✔	●	●	●	○	●	○	○
		10	○	●	✔	✔	✗	✔	○	○	○	○	○	○	○
	Opera	89	●	○	✔	✔	✔	✔	●	●	○	○	●	○	○
		70	●	○	✔	✔	✔	✔	●	●	○	○	●	○	○
		55	○	○	✔	✔	✔	✔	●	●	○	○	●	○	○
	QQ	11	●	○	✔	✔	✔	✔	●	●	○	○	●	○	○
		10.4	○	○	✔	✔	✔	✔	●	●	○	○	●	○	○
		9	○	●	✗	✗	✗	✗	○	○	○	○	●	○	○
	360	13.1	●	○	✔	✔	✔	✔	●	●	○	○	●	○	○
		10	○	○	✔	✔	✔	✔	●	●	○	○	●	○	○
		8.1	○	○	✔	✔	✗	✔	●	●	○	○	●	○	○
Lin.	Chrome	103	●	○	✔	✔	✔	✔	○	●	○	●	●	○	○
	Firefox	102	●	○	✔	✔	✔	✔	●	●	●	○	●	○	○
	Edge	103	●	○	✔	✔	✔	✔	○	●	○	●	●	○	○
	Opera	89	●	○	✔	✔	✔	✔	○	●	○	○	●	○	○
	QQ	70	○	○	✔	✔	✔	✔	○	○	○	○	●	○	○
	360	10.6	●	○	✔	✔	✔	✔	○	●	○	○	●	○	○
Mac.	Chrome	102	●	○	✔	✔	✔	✔	○	●	○	●	●	○	○
	Firefox	102	●	○	✔	✔	✔	✔	●	●	●	○	●	○	○
	Edge	103	●	○	✔	✔	✔	✔	○	●	○	●	●	○	○
	Opera	89	●	○	✔	✔	✔	✔	○	●	○	○	●	○	○
	QQ	4.5	○	○	✗	✗	✗	✗	○	○	○	○	●	○	○
	360	12.2	●	○	✔	✔	✔	✔	○	●	○	○	●	○	○
	Safari	15.1	●	○	✔	✔	✔	✔	○	●	○	●	●	○	○
And.	Chrome	103	●	○	✔	✔	✔	✔	○	○	○	●	●	○	○
	Firefox	102.2	●	○	✔	✔	✔	✔	○	●	○	○	●	○	○
	Edge	103	●	○	✔	✔	✔	✔	○	○	○	●	●	○	○
	Opera	70.3	●	○	✔	✔	✔	✔	○	○	○	○	●	○	○
	QQ	12.9	●	○	✔	✔	✔	✔	○	○	○	○	●	○	○
	360	10	●	○	✔	✔	✔	✔	○	○	○	○	●	○	○
IOS	Chrome	103	●	○	✔	✔	✔	✔	○	●	○	●	●	○	○
	Firefox	102	●	○	✔	✔	✔	✔	○	●	○	○	●	○	○
	Edge	96	●	○	✔	✔	✔	✔	○	●	○	●	●	○	○
	Opera	3.2	●	○	✔	✔	✔	✔	○	○	○	○	●	○	○
	QQ	12.9	●	○	✔	✔	✔	✔	○	●	○	○	○	○	○
	360	4.2	●	○	✔	✔	✔	✔	○	●	○	○	●	○	○
	Safari	15.4	●	○	✔	✔	✔	✔	○	●	○	●	●	○	○

[1] DeTLSVer indicates the deprecated TLS version.
[2] SelfSignCert indicates the self-signed certificate; ExpCert indicates the expired certificate.
[3] NoSANCert indicates the certificate without the Subject Alternative Name field; DomErrCert indicates the certificate's domain name does not match the server.
[4] OCSPStal indicates OCSP Stapling; OCSPMS indicates OCSP Must-Staple.

5 Web Browser Behavior

As a middlebox between Internet users and webmail servers, web browsers play a vital role in authenticating server identities and securing webmail messages. However, it is unclear how well web browsers implement HTTPS-related features and whether it is out of balance with the development of webmail servers. To this end, we comprehensively investigate implementing HTTPS-related features in different versions of eight browsers on five OSes.

5.1 Methodology

To satisfy the need to sign certificates for different test cases, we generate our own root certificate and make the browser trust it. After that, we purchase a domain name and configure different test suites on the Nginx web server. Each test suite is completely independent. Specifically, the TLS version, `HSTS`, and `HPKP` test suite are implemented by changing the Nginx configuration. The certificate error test suite is generated using OpenSSL and signed with our own root certificate. Considering the `OCSP/CRL`, `OCSP Stapling`, and `OCSP Must-Staple` test suites, we build three unique revoked certificates. Specifically, the `CRL` server and `OCSP` server are specified in the revoked certificate of the `OCSP/CRL` test suite. The `OCSP Stapling` test suite prefetches the `OCSP` response file of the revoked certificate and transmits it in the TLS handshake via Nginx. The OID (1.3.6.1.5.5.7.1.24) is added in the revoked certificate extension of the `OCSP Must-Staple` test suite, and we configure the server not to support `OCSP Stapling`. Furthermore, the `DANE-TLSA` test suite is implemented by configuring invalid `TLSA` records and `DNSSEC` for our webmail server. In particular, since some browsers do not perform `CT` on certificates signed by locally trusted CAs [51], we use https://no-sct.badssl.com/ as the `CT` test suite.

Finally, we choose various popular web browsers: Chrome, Firefox, Safari, Microsoft Edge, Internet Explorer, Opera, QQ, and 360 on desktop OSes (Windows 10/7, Ubuntu 20.04, and macOS 12) and mobile OSes (Android 10, and IOS 15.4), with 50 different combinations in total. For each browser, we investigate the configuration of the latest version in different OSes. In particular, we also detect older browser versions in Windows.

5.2 Results

Table 14 shows the implementation of HTTPS-related features of web browsers in different OSes. We find that web browser support for certificate revocation and CT is surprisingly poor. We give details below.

TLS Version. Almost all of the latest versions of browsers support TLSv1.3, but IE, which is Windows' built-in browser, does not. In addition, IE still supports the deprecated TLS version, which undoubtedly brings enormous security risks to webmail. Since Microsoft ultimately terminated support for IE on June 15, 2022 [52], we recommend that users stop using IE as their default browser.

Certificate Error. Most web browsers can detect self-signed certificates, expired certificates, and domain name mismatch errors well. However, IE and older versions of Firefox, QQ, and 360 in Windows allow access to the website without SAN certificates. According to the statistics in Section 4.1, we can infer that at least about 40% of the servers in List-webmail do not work correctly in most browsers due to certificate errors.

Certificate Revocation. Web browsers are pretty poor at implementing certificate revocation mechanisms. Considering CRL and OCSP, almost all OSes (except Windows) browsers do not send certificate status requests to the CRL/OCSP server. Furthermore, only Firefox in desktop OSes respects the OCSP Must-Staple. As for OCSP Stapling, the browsers' support has improved, but we still find some deficiencies. For example, almost all browsers in Android do not support OCSP Stapling.

Additionally, we find that webmail servers and web browser's support for certificate revocation mechanisms are seriously unbalanced. As we can see from Table 10, most webmail servers only support CRL/OCSP, but the browser is the opposite. Therefore, we can conclude that most web browsers cannot effectively detect the revocation status of the webmail server certificate.

CT. The deployment of CT for web browsers is not as widespread as for webmail servers. Specifically, only Chrome, Edge, and Safari detect whether the certificate violates the CT policy. Moreover, these browsers no longer need Expect-CT to enforce CT.

Other. IE 10 in Windows and QQ 12.9 in IOS do not support HSTS. Furthermore, considering HPKP and DANE-TLSA, no browser deploys them.

5.3 Summary

As a critical part of the webmail ecosystem, better HTTPS-related configuration of web browsers can significantly improve webmail security and urge unreliable webmail servers to update. However, we find that browsers vary widely in implementing HTTS-related features, even within the same browser across different OSes. In addition, the configuration of older browser versions is even more worrying. For example, QQ 9.0 hardly detects certificate errors. Therefore, we recommend that Internet users choose the latest browser version whenever possible.

6 Related Work

In this section, we describe several previous studies related to our work, especially in the HTTPS and email ecosystem.

6.1 HTTPS ecosystem

Many works have been devoted to the HTTPS ecosystem from different perspectives, including DANE-TLSA deployment [53], specific HTTPS protocols [34], [54], [55], and the entire HTTPS ecosystem [14], [37], [56], [57]. Among them, the works in [14], [53], [54], [58] are close to our work. Particularly, the first systematic measurement on DANE-TLSA is given in [53], which indicates the corresponding low deployment. Furthermore, the first large-scale measurement of DNSSEC PKI management was performed by Chung et al [58]. They found widespread misconfigurations in the DNSSEC infrastructure. Furthermore, Chung et al. also indicated in [59] that registrar support for DNSSEC is far from expected. At ACM IMC 2015, Kranch et al. [54] conducted the first measurement on HSTS and HPKP. Their results show that the corresponding deployment is quite limited, and there are even many configuration errors in the existing deployments, which severely undermine the security guarantees the protocols provide. Later, Amann et al. [14] at ACM IMC 2017 gave the first comprehensive measurement on many HTTPS security features, including CT, HSTS, HPKP, CAA, TLSA, SCSV, and TLS versions. They showed that lower configuration complexity, better deployment. In particular, SCSV and CT have wider adoption than other features.

Certificates are the root of trust in the TLS and HTTPS ecosystem; hence, numerous studies have been carried out on different segments of the certificate ecosystem, including certificate authority system [60], [61], certificate revocation [50], [62], [63], certificate vulnerability [64]–[67], CT deployment [14], [18], [68]–[70], and CT logs [17], [28], [71]. Like the results obtained in this paper, Chung et al. [50] showed that the proportion of supporting OCSP Must-Staple certificate revocation method was exceptionally rare. In addition, they point out the poor implementation of OCSP Must-Staple by web browsers. Scheitle et al. [68] analyzed the evolution of CT from 2017-04-26 to 2018-05-23. The results show that the certificates in CT logs increased exponentially during that period. Furthermore, 33% of the established connections supported CT at that time.

6.2 Email ecosystem

Many papers analyze the security mechanisms for guaranteeing the security of email, such as TLS [72], DANE-TLSA [26], DKIM (DomainKeys Identified Mail) [73], and SPF (Sender Policy Framework) [74]. However, to the best of our knowledge, only one paper is related to the HTTPS security of webmail services [75]. Particularly, the authors measured the security configuration at each step in the whole process of email delivery, while only two related HTTPS security features (the TLS support and the validity of certificates) were conducted on 22 popular webmail servers. They found that 13.64% of the investigated webmail servers did not support TLS protocol, and all the certificates used in the TLS connections could pass the verification. In contrast, we measured the deployment of more than 10 HTTPS-related security policies and extensions for more than 21k webmail servers. Unlike the result in [75], we found that almost all the investigated webmail servers support TLS protocol. Furthermore, we also found that some HTTPS

TABLE 15: The comparison of scan results.

	Ours		Previous
	List-webmail	List-domain	
HSTS	10.68%	9.28% ↑	3.59% [14]
HPKP	0.01%	0.01% ↓	0.02% [14]
Expect-CT	3.26%	4.19% ↑	0.03% [71]
TLSA	0.16%	0.08% ↑	<0.01% (1246) [14]
CAA	1.37%	1.10% ↑	0.01% [14]
CT	93.68%	93.54% ↑	32.61% [68]
SCSV	92.33%	92.08% ↓	96.80% [14]

features are well-deployed while some are in a bad situation due to the configuration complexity.

7 Discussion

During the past several years, many studies have been devoted to measuring the deployment of security features we investigated in this paper. In the following, we analyze the deployment evolution of HTTPS-related features for email servers and HTTPS servers[2].

As shown in Table 15, we can see that almost all security features have gained better deployment now than that obtained in previous studies. One possible reason for this situation is that more and more organizations pay attention to network security due to security incidents [1], [39]. Another reason may be that all the domains in our experiments are from email addresses, while previous studies contain all kinds of domains. We find that CT has made the most growth among all the security features. We think it is mainly because that Chrome would not trust the new certificate without logging in to CT logs [19]. In contrast, we also find that SCSV and HPKP features have a slight drop compared with previous studies. The former may be due to they do not consider transient errors (e.g., i/o timeout) in their statistics, and the reason for the latter is that Google deprecated HPKP in 2018 due to its complexity [5].

Furthermore, we also find two interesting situations. The former is related to the delivery way of SCT. Amann et al. [14] found that the popular servers were most likely to transmit SCTs via TLS extensions, and they thought that this method aims to save 100 bytes at the beginning of mobile HTTPS connections when clients do not support the CT policy. However, in our scan results, less than 0.01% of servers (only 22) in List-domain and none of the webmail servers (no matter in List-webmail or List-34) transmit SCTs via TLS extensions. This change is mainly because the server should deploy more configurations to enable the SCT transmission via TLS extensions. The latter interesting situation is about the value of max-age. Scheitle et al. [71] found that only 0.03% of servers (7.3k) support Expect-CT, and most of them set the max-age attribute as zero. In contrast, the ratio has grown to 4.19% (473k domains), and more than 95% of these servers set non-zero max-age values. The latter case indicates that the use of the Expect-CT technology has been on the right track.

2. We know that comparing datasets of different sizes and origins can introduce errors in the results. However, we believe our comparison is a meaningful endeavor for understanding the evolution of the relative security status of email servers among HTTPS servers.

8 Limitations and Future Work

We acknowledge that our webmail dataset is not comprehensive enough. In the future, we will collect more email addresses and identify webmail domains more precisely. Moreover, this paper does not consider the security differences between webmail and other email services. To fully understand email security and guide future email development, we plan to compare the security of webmail services and dedicated email services, including dedicated email clients (Outlook, Gmail APP, Apple mail) and corresponding email protocols (Exchange, SMTPS, IMAPS). Finally, our research only analyzes webmail domains protected by HTTPS. Therefore, our future work will further study the effect of QUIC [76] and HTTP/3 [77] on the security and performance of webmail services in different OSes.

9 Conclusion

In this paper, we have provided the first comprehensive security view of the webmail world, including webmail servers and web browsers. Through a detailed analysis of 21k webmail servers and popular top-34 webmail servers extracted from 2.2 million email addresses, we found that some HTTPS-related features gain rare deployments, such as HSTS and OCSP Stapling. Particularly, we have also classified the security levels of webmail servers according to the HTTPS-related feature's protection responsibilities and security benefits. Furthermore, we have investigated HTTPS-related feature implementations for 50 different browser and OS combinations. The results show that browsers have poor certificate revocation and CT implementation, and older browsers have even worse security configurations.

Our results indicate that the lower the deployment complexity, the more adoption. Given the importance of webmail, we hope that server administrators and browser vendors can rebalance the pros and cons of deploying HTTPS-related features.

References

[1] enisa. (2011, December) Operation black tulip: Certificate authorities lose authority. [Online]. Available: https://www.enisa.europa.eu/media/news-items/operation-black-tulip/

[2] J. Hodges, C. Jackson, and A. Barth, "Http strict transport security (hsts)," RFC 6797, November 2012.

[3] C. Evans, C. Palmer, and R. Sleevi, "Public key pinning extension for http," RFC 7469, April 2015.

[4] E. Stark, "Expect-ct extension for http," RFC 9163, June 2020.

[5] J. Medley. (2018) Deprecate http-based public key pinning. [Online]. Available: https://developers.google.com/web/updates/2018/04/chrome-67-deps-rems#deprecate_http-based_public_key_pinning

[6] P. Hoffman and J. Schlyter, "The dns-based authentication of named entities (dane) transport layer security (tls) protocol: Tlsa," RFC 6698, August 2012.

[7] D. Eastlake and C. Kaufman, "Domain name system security extensions," RFC 2065, January 1997.

[8] B. Laurie, A. Langley, and E. Kasper, "Certificate transparency," RFC 6962, June 2013.

[9] B. Moeller and A. Langley, "Tls fallback signaling cipher suite value (scsv) for preventing protocol downgrade attacks," RFC 7507, April 2015.

[10] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile," RFC 5280, May 2008.

[11] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 internet public key infrastructure online certificate status protocol - ocsp," RFC 6960, June 2013.

[12] Y. Pettersen, "The transport layer security (tls) multiple certificate status request extension," RFC 6961, June 2013.

[13] P. Hallam-Baker, "X.509v3 transport layer security (tls) feature extension," RFC 7633, October 2015.

[14] J. Amann, O. Gasser, Q. Scheitle, L. Brent, G. Carle, and R. Holz, "Mission accomplished?: HTTPS security after diginotar," in *ACM IMC*, 2017, pp. 325–340.

[15] M. Kranch and J. Bonneau, "Upgrading HTTPS in mid-air: An empirical study of strict transport security and key pinning," in *NDSS*, 2015.

[16] P. Szalachowski and A. Perrig, "Short paper: On deployment of dns-based security enhancements," in *FC*, 2017, pp. 424–433.

[17] J. Gustafsson, G. Overier, M. F. Arlitt, and N. Carlsson, "A first look at the CT landscape: Certificate transparency logs in practice," in *PAM*, 2017, pp. 87–99.

[18] C. Nykvist, L. Sjöström, J. Gustafsson, and N. Carlsson, "Server-side adoption of certificate transparency," in *PAM*, 2018, pp. 186–199.

[19] R. Sleevi. (2017, April) Certificate transparency in chrome - change to enforcement date. [Online]. Available: https://groups.google.com/a/chromium.org/g/ct-policy/c/sz_3W_xKBNY/m/6jq2ghJXBAAJ

[20] Apple. (2021, March) Apple's certificate transparency policy. [Online]. Available: https://support.apple.com/en-us/HT205280

[21] E. Rescorla, "The transport layer security (tls) protocol version 1.3," RFC 8446, August 2018.

[22] Have i been pawned? [Online]. Available: https://haveibeenpwned.com/

[23] S. L. Tech, J. Martinson, mohityadav04, and M. Rofail, "gocolly," https://github.com/gocolly/colly, 2021.

[24] F. Labs. Web filter lookup. [Online]. Available: https://www.fortiguard.com/webfilter

[25] NetSTAR. Url/ip lookup. [Online]. Available: https://incompass-branch.netstar-inc.com/urlsearch

[26] H. Lee, A. Gireesh, R. van Rijswijk-Deij, T. Kwon, and T. Chung, "A longitudinal and comprehensive study of the DANE ecosystem in email," in *USENIX Security*, 2020, pp. 613–630.

[27] S. . Data. Most popular email providers in history. [Online]. Available: https://statisticsanddata.org/data/most-popular-email-providers-in-history/

[28] B. VanderSloot, J. Amann, M. Bernhard, Z. Durumeric, M. Bailey, and J. A. Halderman, "Towards a complete view of the certificate ecosystem," in *ACM IMC*, 2016, pp. 543–549.

[29] D. Eastlake, "Transport layer security (tls) extensions: Extension definitions," RFC 6066, January 2011.

[30] Q. Scheitle, "massdns," https://github.com/quirins/massdns, 2018.

[31] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast internet-wide scanning and its security applications," in *USENIX Security Symposium*, 2013, pp. 605–620.

[32] mkcert. Who do you trust? [Online]. Available: https://mkcert.org/

[33] O. Gasser and R. Holz, "goscanner," https://github.com/tls-evolution/goscanner, 2019.

[34] R. Holz, J. Hiller, J. Amann, A. Razaghpanah, T. Jost, N. Vallina-Rodriguez, and O. Hohlfeld, "Tracking the deployment of TLS 1.3 on the web: a story of experimentation and centralization," *Computer Communication Review*, pp. 3–15, 2020.

[35] N. Labs. (2021, February) Unbound 1.13.1 dns resolver. [Online]. Available: https://www.unbound.net

[36] Chrome. Chrome's log-list. [Online]. Available: https://www.gstatic.com/ct/log_list/v2/log_list.json

[37] P. Kotzias, A. Razaghpanah, J. Amann, K. G. Paterson, N. Vallina-Rodriguez, and J. Caballero, "Coming of age: A longitudinal study of TLS deployment," in *ACM IMC*, 2018, pp. 415–428.

[38] Wikipedia. Protonmail. [Online]. Available: https://zh.wikipedia.org/wiki/ProtonMail

[39] M. Marlinspike, "New tricks for defeating ssl in practice," *Black Hat DC*, vol. 2, 2009.

[40] Wikipedia. Daum. [Online]. Available: https://zh.wikipedia.org/wiki/Daum

[41] Chromium. Chromium preload list. [Online]. Available: https://cs.chromium.org/chromium/src/net/http/transport_security_state_static.json

[42] Mozilla. Preloading hsts. [Online]. Available: https://blog.mozilla.org/security/2012/11/01/preloading-hsts/

[43] M. E. Team. (2015, June) Http strict transport security comes to internet explorer 11 on windows 8.1 and windows 7. [Online]. Available: https://blogs.windows.com/msedgedev/2015/06/09/http-strict-transport-security-comes-to-internet-explorer-11-on-windows-8-1-and-windows-7/

[44] Mozilla. Hpkp preload list. [Online]. Available: https://wiki.mozilla.org/SecurityEngineering/Public_Key_Pinning/Implementation_Details

[45] GoogleChrome. Chrome certificate transparency policy. [Online]. Available: https://github.com/GoogleChrome/CertificateTransparency/blob/master/ct_policy.md

[46] B. Laurie. (2015, May) Improving the security of ev certificates. [Online]. Available: https://sites.google.com/site/certificatetransparency/ev-ct-plan

[47] S. S. Team. (2020, January) Dv, ov, iv, and ev certificates. [Online]. Available: https://www.ssl.com/ARTICLE/dv-ov-and-ev-certificates/

[48] Mozilla. (2019, October) Improved security and privacy indicators in firefox 70. [Online]. Available: https://blog.mozilla.org/security/2019/10/15/improved-security-and-privacy-indicators-in-firefox-70/

[49] T. Hunt. (2018, September) Extended validation certificates are dead. [Online]. Available: https://www.troyhunt.com/extended-validation-certificates-are-dead/

[50] T. Chung, J. Lok, B. Chandrasekaran, D. R. Choffnes, D. Levin, B. M. Maggs, A. Mislove, J. P. Rula, N. Sullivan, and C. Wilson, "Is the web ready for OCSP must-staple?" in *ACM IMC*, 2018, pp. 105–118.

[51] Google. Certificate transparency. [Online]. Available: https://chromium.googlesource.com/chromium/src/+/master/net/docs/certificate-transparency.md

[52] Microsoft. (2022, June) Internet explorer 11 has retired and is officially out of support. [Online]. Available: https://blogs.windows.com/windowsexperience/2022/06/15/internet-explorer-11-has-retired-and-is-officially-out-of-support-what-you-need-to-know/

[53] L. Zhu, D. Wessels, A. Mankin, and J. Heidemann, "Measuring dane tlsa deployment," in *TMA*, 2015, pp. 219–232.

[54] M. Kranch and J. Bonneau, "Upgrading HTTPS in mid-air: An empirical study of strict transport security and key pinning," in *The Internet Society NDSS*, 2015.

[55] S. de los Santos, C. Torrano, Y. Rubio, and F. Brezo, "Implementation state of HSTS and HPKP in both browsers and servers," in *CANS*, 2016, pp. 192–207.

[56] O. Levillain, "A study of the TLS ecosystem. (une étude de l'écosystème TLS)," Ph.D. dissertation, Telecom & Management SudParis, France, 2016.

[57] A. P. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz, "Measuring HTTPS adoption on the web," in *USENIX Security*, 2017, pp. 1323–1338.

[58] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "A longitudinal, end-to-end view of the dnssec ecosystem," in *USENIX Security 17*, 2017, pp. 1307–1322.

[59] T. Chung, R. van Rijswijk-Deij, D. R. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "Understanding the role of registrars in DNSSEC deployment," in *IMC*. ACM, 2017, pp. 369–383.

[60] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman, "Analysis of the HTTPS certificate ecosystem," in *ACM IMC*, K. Papagiannaki, P. K. Gummadi, and C. Partridge, Eds., 2013, pp. 291–304.

[61] M. Aertsen, M. Korczynski, G. C. M. Moura, S. Tajalizadehkhoob, and J. van den Berg, "No domain left behind: is let's encrypt democratizing encryption?" in *ACM ANRW*, 2017, pp. 48–54.

[62] L. Zhang, D. R. Choffnes, D. Levin, T. Dumitras, A. Mislove, A. Schulman, and C. Wilson, "Analysis of SSL certificate reissues and revocations in the wake of heartbleed," in *ACM IMC*, 2014, pp. 489–502.

[63] L. Zhu, J. Amann, and J. S. Heidemann, "Measuring the latency and pervasiveness of TLS certificate revocation," in *PAM*, 2016, pp. 16–29.

[64] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer, and V. Paxson, "The matter of heartbleed," in *ACM IMC*, 2014, pp. 475–488.

[65] M. E. Acer, E. Stark, A. P. Felt, S. Fahl, R. Bhargava, B. Dev, M. Braithwaite, R. Sleevi, and P. Tabriz, "Where the wild warnings are: Root causes of chrome HTTPS certificate errors," in *ACM CCS*, 2017, pp. 1407–1420.
[66] D. Kumar, Z. Wang, M. Hyder, J. Dickinson, G. Beck, D. Adrian, J. Mason, Z. Durumeric, J. A. Halderman, and M. Bailey, "Tracking certificate misissuance in the wild," in *IEEE S&P*, 2018, pp. 785–798.
[67] M. Cui, Z. Cao, and G. Xiong, "How is the forged certificates in the wild: Practice on large-scale SSL usage measurement and analysis," in *ICCS*, 2018, pp. 654–667.
[68] Q. Scheitle, O. Gasser, T. Nolte, J. Amann, L. Brent, G. Carle, R. Holz, T. C. Schmidt, and M. Wählisch, "The rise of certificate transparency and its implications on the internet ecosystem," in *ACM IMC*, 2018, pp. 343–349.
[69] N. Blagov and M. Helm, "State of the certificate transparency ecosystem," *Network*, 2020.
[70] B. Li, F. Li, Z. Ma, and Q. Wu, "Exploring the security of certificate transparency in the wild," in *ACNS*, 2020, pp. 453–470.
[71] O. Gasser, B. Hof, M. Helm, M. Korczynski, R. Holz, and G. Carle, "In log we trust: Revealing poor security practices with certificate transparency logs and internet measurements," in *PAM*, 2018, pp. 173–185.
[72] R. Holz, J. Amann, O. Mehani, M. A. Kâafar, and M. Wachs, "TLS in the wild: An internet-wide analysis of tls-based protocols for electronic communication," in *NDSS*, 2016.
[73] C. Wang, K. Shen, M. Guo, Y. Zhao, M. Zhang, J. Chen, B. Liu, X. Zheng, H. Duan, Y. Lin *et al.*, "A large-scale and longitudinal measurement study of dkim deployment."
[74] Z. Durumeric, D. Adrian, A. Mirian, J. Kasten, E. Bursztein, N. Lidzborski, K. Thomas, V. Eranti, M. Bailey, and J. A. Halderman, "Neither snow nor rain nor MITM...: an empirical analysis of email delivery security," in *ACM IMC*, 2015, pp. 27–39.
[75] I. D. Foster, J. Larson, M. Masich, A. C. Snoeren, S. Savage, and K. Levchenko, "Security by any other name: On the effectiveness of provider based email security," in *ACM CCS*, 2015, pp. 450–464.
[76] J. Iyengar and M. Thomson, "Quic: A udp-based multiplexed and secure transport," RFC 9000, May 2021.
[77] M. Bishop, "Http/3," RFC 9114, June 2022.

Ruixuan Li is a master student of the School of Computer and Information Engineering at Zhejiang Gongshang University. His research interests include internet security and network measurement.

Zhenyong Zhang is a master student of the School of Computer and Information Engineering at Zhejiang Gongshang University. His research interests include internet security and network monitoring.

Jun Shao (M'21-SM'22) received the Ph.D. degree from the Department of Computer Science and Engineering at Shanghai Jiao Tong University, Shanghai, China in 2008. He was a postdoc in the School of Information Sciences and Technology at Pennsylvania State University, USA from May 2008 to April 2010. He was also a visiting professor in the Faculty of Computer Science, University of New Brunswick, Canada from October 2017 to March 2018. He is currently a professor of the School of Computer and Information Engineering at Zhejiang Gongshang University, Hangzhou, China. His research interests include network security and applied cryptography.

Rongxing Lu (S'09-M'11-SM'15-F'21) is an associate professor at the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Canada. Before that, he worked as an assistant professor at the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore from April 2013 to August 2016. Rongxing Lu worked as a Postdoctoral Fellow at the University of Waterloo from May 2012 to April 2013. He was awarded the most prestigious "Governor General's Gold Medal", when he received his PhD degree from the Department of Electrical & Computer Engineering, University of Waterloo, Canada, in 2012; and won the 8th IEEE Communications Society (ComSoc) Asia Pacific (AP) Outstanding Young Researcher Award, in 2013. Also, Dr. Lu received his first PhD degree at Shanghai Jiao Tong University, China, in 2006. Dr. Lu is an IEEE Fellow. His research interests include applied cryptography, privacy enhancing technologies, and IoT-Big Data security and privacy. He has published extensively in his areas of expertise (with H-index 72 from Google Scholar as of November 2020), and was the recipient of 9 best (student) paper awards from some reputable journals and conferences. Currently, Dr. Lu serves as the Vice-Chair (Conferences) of IEEE ComSoc CIS-TC (Communications and Information Security Technical Committee). Dr. Lu is the Winner of 2016-17 Excellence in Teaching Award, FCS, UNB.

Xiaoqi Jia Xiaoqi Jia received his Ph.D. degree from Chinese Academy of Sciences, China in 2010. He is currently a Professor at the Institute of Information Engineering, Chinese Academy of Sciences, China. His main research interests include operating system security, cloud security, and virtualization technologies.

Guiyi Wei is a professor of the School of Information and Electronic Engineering at Zhejiang Gongshang University. He obtained his Ph.D. in Dec 2006 from Zhejiang University, where he was advised by Cheung Kong chair professor Yao Zheng. His research interests include wireless networks, mobile computing, cloud computing, social networks and network security.