A Large-scale and Longitudinal Measurement Study of DKIM Deployment

Chuhan Wang [†] *Kaiwen Shen* [†] *Minglei Guo* [†] *Yuxuan Zhao* [‡] *Mingming Zhang* [†]
Jianjun Chen [†✉] *Baojun Liu* [†✉] *Xiaofeng Zheng* [†§] *Haixin Duan* [†§] *Yanzhong Lin* [¶] *Qingfeng Pan* [¶]
[†]*Tsinghua University*
[‡]*North China Institute of Computing Technology*
[§]*Qi An Xin Technology Research Institute*
[¶]*Coremail Technology Co. Ltd*

Abstract

DomainKeys Identified Mail (DKIM) is an email authentication protocol to protect the integrity of email contents. It has been proposed and standardized for over a decade and adopted by Yahoo!, Google, and other leading email service providers. However, little has been done to understand the adoption rate and potential security issues of DKIM due to the challenges of measuring DKIM deployment at scale.

In this paper, we provide a large-scale and longitudinal measurement study on how well DKIM is deployed and managed. Our study was made possible by a broad collection of datasets, including 9.5 million DKIM records from passive DNS datasets over five years and 460 million DKIM signatures from real-world email headers. Moreover, we conduct an active measurement on Alexa Top 1 million domains. Our measurement results show that 28.1% of Alexa Top 1 million domains have enabled DKIM, of which 2.9% are misconfigured. We demonstrate that the issues of DKIM key management and DKIM signatures are prevalent in the real world, even for well-known email providers (e.g., Gmail and Mail.ru). We recommend the security community should pay more attention to the systemic problems of DKIM deployment and mitigate these issues from the perspective of protocol design.

1 Introduction

Since the Simple Mail Transfer Protocol (SMTP) [28] lacks authentication mechanism [11, 18, 31], email services have long been fraught with email spoofing attacks [2, 5, 7]. To address this security issue, the Internet Engineering Task Force (IETF) has proposed three email authentication protocols, including Sender Policy Framework (SPF) [19], DomainKeys Identified Mail (DKIM) [13], and Domain-based Message Authentication, Reporting, and Conformance (DMARC) [21]. These protocols protect user identities in different ways, and they need to cooperate to protect email authentication.

✉ Corresponding authors:{jianjun, lbj}@tsinghua.edu.cn.

DKIM is an essential part of the email authentication chain. It relies on digital signatures to prevent emails from being forged or tampered with. Unlike the other two protocols, DKIM focuses on verifying the integrity of email contents. However, the complexity of DKIM deployment creates multiple management issues. Previous studies have shown that DKIM misconfigurations may allow adversaries to successfully send spoofing emails that can bypass both DKIM and DMARC verification [11, 31]. Besides, similar to other security mechanisms based on cryptography, such as DNSSEC and TLS, DKIM may be prone to common key management risks. Therefore, it is significant to understand the current status of DKIM deployment to improve the protocol design, implementation, and management practices.

While significant efforts have been devoted to SPF and DMARC deployment [14, 15, 18], the deployment of DKIM has been paid less attention in the email ecosystem, since it is challenging to obtain DKIM records and measure DKIM deployment further. Intuitively, there are two kinds of methods to extract DKIM information: actively scanning DKIM public keys or passively extracting DKIM signatures from email headers. Unfortunately, active scanning method is not suitable for measuring DKIM public keys. DKIM public keys are published through the DNS `TXT` records as "`selector._domainkey.example.com`". The `selector` field is chosen randomly by domain owners and thus unpredictable. Besides, DKIM signatures are also difficult to collect by security researchers. DKIM signatures are embedded in the email header and would not be publicly released. It is difficult for researchers to obtain large amounts of DKIM signature data for further security analysis.

In this paper, we perform the first large-scale and longitudinal measurement study on the current status of DKIM deployment by both passive analysis and active scanning, to the authors' best knowledge. First, we extract DKIM records from two passive DNS datasets and obtain DKIM signatures in email headers by cooperating with our industry partners. Our collected DKIM records covering 5 million domain names and 2 million DKIM selectors and spanning more

than five years. Then, leveraging the popular DKIM selectors investigated from our passive datasets, we actively and heuristically query DKIM records for Alexa Top 1 million domains, and find at least 28.1% of the domains have deployed DKIM. DKIM deployment rates vary significantly with different TLDs: the domain names under *.edu* show the highest rate (71.3%) in the tested generic Top Level Domains (gTLDs), and the highest rate (58.6%) in country code Top Level Domains (ccTLDs) comes from *.au* (Australia) domains.

While investigating DKIM mismanagement, we find prevalent security issues in the real world, even for the leading email service providers like Yahoo and Gmail. First of all, within Alexa Top 1 million domains, we find 8,147 deployment records that can not be validated due to missing or incorrect DKIM records, accounting for 2.9% of all DKIM-enabled domains. Even worse, 3,292 domains were configured with abnormal DKIM fields, resulting in parsing errors of the corresponding public keys. Secondly, our research demonstrates that DKIM key management issues are prevalent: 66.9% of DKIM-enabled domains adopt shared DKIM keys, and 84% use weak DKIM keys. Besides, 8.4% of domains have not rotated DKIM keys in the past five years.

We show that 94.2% of the domains in our dataset have DKIM signature issues. 94.1% of domains use weak DKIM signatures without including necessary email headers, such as `From`, `To`, `Subject`, `Content-Type`, `Reply-To`, `Date`, and `Cc`. And, only 2.2% domains have deployed the oversigning protection mechanism that is recommended by the RFC 6376 [13]. Our results show that 6,860 (0.3%) domains still use the "`l=`" tag in the DKIM signatures, which can display fraudulent content to end-users without breaking DKIM signatures, and 65.9% of domains still use the outdated hash algorithm, i.e., SHA-1.

We have tried our best to contact the affected email providers and report these issues, and developed an online testing tool to help email administrators verify and deploy their DKIM records. We believe that the online tool is helpful for enhancing DKIM deployment.

Contributions. The contributions of the paper are as follows:

- We perform the first large-scale and longitudinal measurement study on the deployment and management of DKIM and find 28.1% of Alexa top 1 million domains have enabled DKIM protection, of which 2.9% are mismanaged.
- We discover that DKIM key management and DKIM signature issues are prevalent in the real world.
- We report the vulnerabilities to the affected email providers and provide an online DKIM testing tool to improve the security of DKIM deployment.

2 Background

The IETF has developed various standard protocols to protect email services from spoofing attacks, including SPF, DKIM, and DMARC. It is necessary to understand these protocols and their cooperation in authenticating email-sender identities.

2.1 DKIM

DomainKeys Identified Mail (DKIM) [13], as an essential email authentication protocol, provides integrity and authenticity protection for email transmission and is used to defend against spoofing and phishing attacks [17].

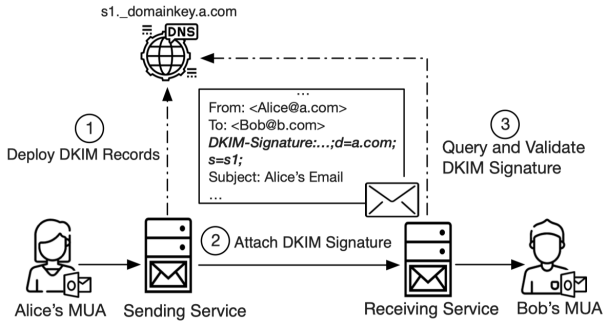

Figure 1: DKIM Verification Workflow.

DKIM Workflow. Figure 1 shows the three steps of DKIM workflow. (1) A domain owner first generates a pair of keys and publishes the public key (DKIM record) via a DNS TXT record. For example, if the owner of a.com sets `s1` as its selector, the DKIM record is set through a TXT record (like the one in Figure 3) of `s1._domainkey.a.com`. (2) The sending service calculates three hashes: the hash of the email body (body-hash), the hash of selected email headers (h-headers), and the hash of the whole email (data-hash). The data-hash consists of body-hash, h-headers, and the DKIM-Signature header with the exception of the value portion of the "`b=`" tag. Then, the email service will sign the data-hash and set the result as the value of the "`b=`" tag. After that, the email service will insert the DKIM signature in the header block and send the email to receivers. (3) The receiving service retrieves the public key from the sender's DNS server to validate the DKIM signature on receiving the email.

DKIM Signatures and DKIM Records. DKIM signature headers and DKIM DNS records consist of different informational elements that are represented by multiple `tag=value` pairs. The examples of a DKIM signature header and a DKIM record published on the DNS server are shown in Figure 2 and Figure 3. We mainly parse the tags from our datasets to further analyze DKIM configurations. The important tags we have analyzed in DKIM records include:

- `k` represents the key type. The default type is "rsa", and RFC 8463 [24] adds "ed25519 " to the key type to support the Edwards-Curve Digital Signature Algorithm.
- `p` represents the public key. An empty value means the public key has been revoked. This tag value is defined by the "k=" tag before being encoded in base64.
- `h` represents acceptable hash algorithms. This tag is optional and the default allows all algorithms (e.g., SHA-1 and SHA-256).

```
DKIM-Signature: v=1; a=rsa-sha256;
c=relaxed/relaxed; s=s1; d=a.com; h=From:
To: Subject; l=200; bh=vYFvy46eesudgj4s...;
b=IHEFQ+7rcisqsRBSEdd83...
```

Figure 2: DKIM Signature Example in Email Headers.

```
v=DKIM1; k=rsa; h=sha256;
p=MIGfMA0GCSqGSIb3DQCyOmR3diPVt1...
```

Figure 3: DKIM Record Example Published on DNS Servers.

The important tags we have analyzed in DKIM signatures include:

- `a` represents the algorithm (e.g., "RSA-SHA1", "RSA-SHA256") for generating the DKIM signature. RFC 6376 [13] recommends that signers should sign using"RSA-SHA256".
- `c` represents the message canonicalization algorithm. It consists of two names separated by a "slash" character, corresponding to the header and body canonicalization algorithms, respectively. Default is "simple/simple". There are two kinds of canonicalization algorithms. The "simple" algorithm tolerates almost no modification, and the "relaxed" algorithm tolerates common modifications such as whitespace replacement and header field line rewrapping.
- `s` represents the selector, an attribute in the DKIM signature, which permits multiple keys under the same domain. Email receivers use this tag to obtain the public key by querying *s1._domainkey.a.com*.
- `d` represents the signer's domain.
- `h` represents the list of headers protected by the signature. `From` header must be included.
- `l` is an optional tag indicating the number of bytes of the email body covered by the signature.
- `bh` is the hash of the canonicalized body part of the message as limited by the "l=" tag.
- `b` represents the actual digital signature of the whole email message, including the email body and the email headers.

2.2 SPF and DMARC

Sender Policy Framework (SPF) [19] and Domain-based Message Authentication, Reporting and Conformance (DMARC) [21] are crucial mechanisms for email sender authentication.

SPF allows a domain owner to publish DNS records to specify which email servers can send emails representing their domain. When receiving an email, the receiving email services can leverage the IP address range from the DNS records to check whether the sending email server is legal. In this way, SPF provides spoofing protection by limiting the sender's IP addresses.

DMARC is an authentication system based on the results of SPF and DKIM verification. It enables the domain owner to publish a policy to specify what actions the receiver should take when the incoming email fails in the DMARC check. When receiving an email, the receiving email services do the SPF and DKIM check first. If the email passes one of the two protocols, then they perform an identifier alignment test to check whether the domain in the `From` header matches the domain name verified by SPF or DKIM.

3 Dataset and Methodology

To investigate the current deployment of DKIM, we mainly follow the three steps shown in Figure 4, including data collection, processing, and analysis.

3.1 Data Collection

It is almost impossible to collect DKIM dataset at scale without knowing the selector for each domain. However, we find two ways to get DKIM information: (1) parsing DKIM records from Passive DNS datasets and (2) extracting DKIM signatures from email headers. Thus, we collaborate with our industrial partners and get the datasets shown in Table 1, including (1) passive DNS datasets from Chinese top security providers, Qi-Anxin and 360, and (2) DKIM signatures from Coremail, the leading email service provider in China.

Passive DNS. When receiving an email with a DKIM signature, the DKIM-enabled email server will retrieve the public key to verify the DKIM signature. Therefore, the DKIM records can be recorded in the DNS traffic of the DNS servers used by receiving email servers. In this research, we use two passive DNS datasets (similar to Farsight DNSDB [4]), 360

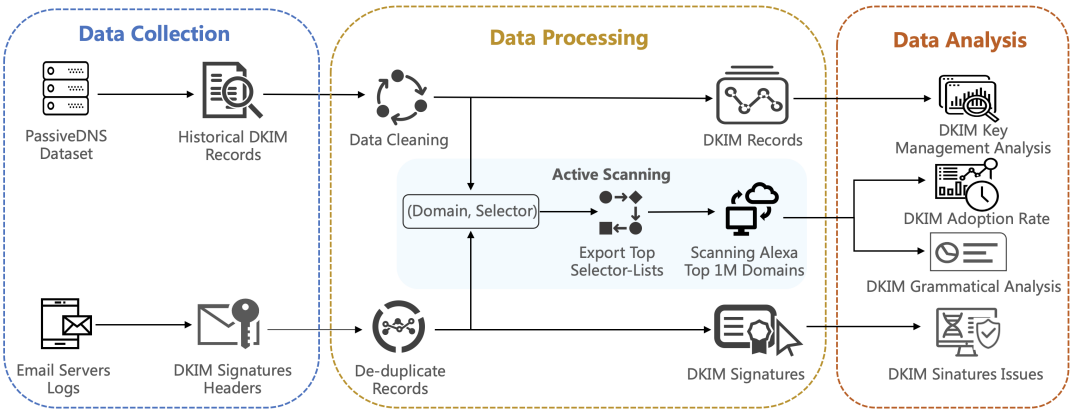

Figure 4: Overview of DKIM Data Collection and Analysis.

PassiveDNS[1] and QiAnXin PassiveDNS[2]. These datasets are extracted from logs of the most popular public DNS resolvers in China, such as 114.114.114.114 (like Google's 8.8.8.8). According to our partners, DNS queries to these public DNS resolvers account for about 10% of all DNS queries in China. Our datasets cover queries from Jun. 2015 to Nov. 2020.

DKIM public keys are stored in DNS TXT records of domains in the same pattern, which is `<selector>._domainkey.<domain>`. Thus, we extract all DKIM records from the passive DNS dataset by matching the "`._domainkey.`" pattern in domain names. The datasets not only contain the DKIM deployment status of a large number of domain names but also record the changes in the DKIM deployment status over time, which allow us to analyze the mismanagement of DKIM from past to present.

Email Server Log. Besides the passive DNS traffic, the email server log is another source to get DKIM information. In this study, we extract 464 million DKIM signatures from the real-world email headers collected by Coremail. These signatures are collected spanning from Mar. 20, 2020, to Oct. 19, 2020, as shown in Table 1. We can parse the `domains` and the relevant `selectors` from DKIM signature headers collected in the email server log, then use them to look up DKIM records via DNS. The dataset provides us with another perspective to analyze DKIM deployment from DKIM signatures.

3.2 Data Processing

Data Cleaning for Passive DNS Data. The passive DNS datasets can not be directly used for analysis, including many wildcard DNS records and misconfigurations. For example, domains can configure their SPF and DMARC records as wildcard DNS records to protect their subdomains. These records can impact our experiment statistics since we need to show the DKIM deployment status on the popular domain names. Thus, we need to clean the passive DNS data before analyzing it.

[1]https://passivedns.cn/help/
[2]https://secrank.cn/passivedns

Table 1: Overview of DKIM Datasets.

	PassiveDNS[1]	DKIM Signature[2]	Total
Distinct Selectors	2,179,653	314,767	2,376,077
Distinct Domains	3,627,871	2,203,628	5,444,288
Alexa Top-1M	87,292	53,302	101,934

[1] Historical DKIM records from Qianxin PassiveDNS and 360 PassiveDNS between Jun. 2015 to Nov. 2020. Qianxin and 360 are the two largest listed cyber security companies in China.
[2] DKIM signature headers from Coremail's email server log between Mar. 2020 to Oct. 2020. Coremail is one of the famous email providers in China.

First, we aggregate the passive DNS records, represented as a 5-tuple (the requested domain name, DKIM record content, the first request timestamp, the last request timestamp, and request times), by domain names. If the DKIM records of a domain name are the same, we merge the 5-tuple records by extending timestamps and request times. If the DKIM records are changed, we treat them as two different 5-tuple records.

Second, we develop a *grammar parser* to analyze the DKIM records in the PassiveDNS data according to RFC 6376 [13], and use it to filter out the DKIM records that violate the standard. The invalid DKIM records are discussed in Section 5. Table 1 presents the statistics of our parsed DKIM records. The Passive DNS datasets include 3.6 million unique domain names with valid DKIM records, covering 87,292 domain names within Alexa Top 1M domains.

Records Deduplication for Email Server Logs. The real-time email server log contains plenty of emails from the same domain name. However, we only focus on the diversity of the domain names and their related DKIM signatures rather than the number of emails. Besides, DKIM signatures with the same domain and selector can share the same configuration. So we de-duplicate these DKIM signatures according to the "d=" tag (domain) and the "s=" tag (selector). Finally, we get 2,252,528 distinct DKIM signatures, including 2,203,628 dis-

Table 2: Top 10 Popular Selectors.

Rank	Selector Name	# Domain	%
1	mail	643,940	11.8%
2	tvdnhvr	481,768	8.9%
3	default	457,069	8.4%
4	zplfznz	391,766	7.2%
5	20150623	384,472	7.1%
6	dkim	190,637	3.5%
7	k1	69,385	1.3%
8	google	62,148	1.1%
9	selector2	34,187	0.6%
10	key1	25,034	0.5%

tinct domain names, among which 53,302 are ranked within Alexa Top 1 Million.

Active Scanning for the Alexa Top 1 Million Domains. Although our passive datasets have a large number of records, they can not cover all popular domain names. To know the DKIM deployment status for the most popular domain names, we also start an active scanning process. For each domain name covered by the active scanning, we need to know its `selector`, and then we can get its DKIM record by accessing its DNS server.

Based on the passive datasets, we can get the mapping of selectors and domains, then further de-duplicate and count the popular selectors. We find the most popular selector "`mail`" is used by more than 643 thousand domain names, which accounts for 11.83% in our datasets. Table 2 shows the top 10 selectors in our datasets, including the common labels like "`default`", "`mail`" and "`dkim`". All of these selectors are used by at least 10k domain names. This feature gives us the opportunity to conduct an active scanning on the DKIM deployment.

After we collected popular selectors, we used them to look up the corresponding DKIM public key records for Alexa top domain names. The overall query volume in our measurement is enormous because it depends on the Cartesian product of the domain name list and the selector list. Using too many selectors will lead to higher time overhead and influence the related DNS services. According to our test on Alexa top 10,000 domains, we find the growth of the newly discovered DKIM domain names becomes slow when using more than 40 selectors to measure, as shown in Figure 5. Thus, we select the top 40 selectors to actively scan the Alexa top 1 million domains. Results show that 28.1% of the domain names have enabled DKIM. We will introduce more details about DKIM adoption in Section 4.

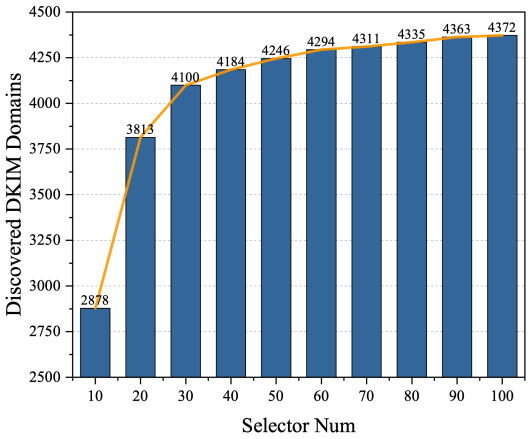

Figure 5: Discovered DKIM Domain Numbers when Using Different Number of Selectors.

3.3 Data Analysis

Based on the collected DKIM data, we are able to draw a big picture of DKIM deployment status from the following four aspects. Here we give an overview of these aspects, and the results will be further introduced and discussed in Section 4-7 in detail.

Adoption Rate of DKIM. We combine the passive collection data and the results of active scanning, and finally calculate the DKIM deployment rate among Alexa Top 1M domains. To know more about DKIM deployment, we also analyze the DKIM adoption of the domains under different ccTLD and gTLD.

Grammatical Analysis of DKIM Records. We filter the measurement results of Alexa Top 1M domains and use the *grammar parser* mentioned in the data processing step (Section 3.2) to analyze the DKIM records. Then, we group all abnormal DKIM records into five categories by their error types.

Management Issues of DKIM Keys. Since our passive DNS data contains the timestamps of the DKIM records, we can analyze the lifetime of these DKIM records. In this part, we extract the public keys from the DKIM records and analyze the issues of DKIM keys from three aspects: (1) the lifetime of the DKIM keys, (2) the sharing of the DKIM keys, and (3) the length of the DKIM keys.

Security Issues of DKIM Signatures. The DKIM signature datasets provide us with a new perspective to analyze DKIM deployment issues. We analyze security issues of DKIM signatures from two aspects: (1) whether the DKIM signatures cover the email headers that are important for security, and (2) whether the obsolete "l=" tag is used. (3) whether the outdated hash algorithm (SHA-1) is used.

3.4 Ethical Considerations

Data Collection. Our passive DNS dataset does not include any privacy data, as was used in previous researches [9, 16]. As for the DKIM signatures, we can only get and use the `DKIM-Signature` headers provided by Coremail. Before that, any users' private information, like email addresses and email bodies, has already been removed. Besides, the DKIM signatures are stored in Coremail's server, and we only access and analyze the data through their bastion host.

Active Scanning. We deploy the scanning tool on ten nodes worldwide and control the frequency interval of active scanning. Only about 40 DNS queries will be performed for each domain name, and the interval between the query for the same domain will be at least five minutes, which will not affect the normal DNS services of the involved domains.

4 Adoption Rate of DKIM

To understand the current adoption of DKIM, we conduct a measurement on the DKIM deployment among *popular email providers* and *Alexa Top 1M domains*.

4.1 Popular Email Providers

For selecting email providers, we first check the email services analyzed by Hu et al. [18] and filter out the ones that can not be used normally in China (e.g., gmx.com and tutanota.com). We also supplement the provider list via Google search. In total, we have investigated the DKIM deployment of 24 popular email providers (Table 3).

We evaluate these email providers from both the sender side and the receiver side. From the sender side, we use each email provider to send emails and check whether DKIM signatures are included. From the receiver side, we send emails with normal DKIM signatures from our domain to each email provider and check DNS request records to judge whether DKIM verification is performed.

The results show that most email providers have adopted DKIM. For the 24 tested email providers, we find all of them conduct DKIM verification from the receiver side, and 6 of them have not deployed DKIM from the sender side.

4.2 Alexa Top 1 Million Domains

Besides popular email providers, we also measure the DKIM deployment among Alexa top 1 million domains. Note that DKIM can achieve a better effect of preventing email spoofing attacks, together with SPF and DMARC. Thus, we also measure the deployment rate of SPF and DMARC by active scanning, and the results can reflect the current status of DKIM deployment from the side.

An overview of the DKIM Adoption Rate among Alexa Top 1M Domains. We find 28.1% domains support DKIM based on active scanning on Alexa top 1M domains (Table 4), and the DKIM records obtained from our passive datasets (101,855 domains). The adoption rate of DKIM is between that of DMARC (11.9%) and SPF (54.1%), which is in line with expectations because SPF is proposed earlier than DKIM, and the design of DMARC is based on SPF and DKIM. Among the Alexa top 1 million domains, 748,993 domains have MX records, of which the adoption rate of security protocols is relatively higher (SPF 69.8%, DKIM 37.0%, DMARC 15.1%).

Table 3: DKIM Adoption of Popular Email Providers.

Email Provider	DKIM	Email Provider	DKIM
gmail.com	✓	yeah.net	✓
mail.ru	✓	126.com	✓
zohu.com	✓	139.com	✗
icloud.com	✓	tom.com	✓
yahoo.com	✓	21cn.com	✗
outlook.com	✓	rambler.ru	✓
yandex.com	✓	cock.li	✓
aol.com	✓	onet.pl	✗
qq.com	✓	runbox.com	✓
sina.com	✗	freemail.hu	✓
sohu.com	✗	naver.com	✓
163.com	✓	daum.net	✗

Table 4: SPF/DKIM/DMARC Adoption Rate among Alexa Top 1 Million Domains.

	# All Domains (%)	# MX Domains (%)
Alexa List	1,000,000 (100.0%)	748,993 (100.0%)
w/ SPF	541,008 (54.1%)	522,696 (69.8%)
w/ DKIM	280,786 (28.1%)	276,827 (37.0%)
w/ DMARC	118,468 (11.9%)	112,798 (15.1%)

Comparing the Adoption Rate by Different TLDs. Since DKIM is deployed for email service domains, we only analyze MX domains in the following. We first focus on generic top-level domains (gTLDs) such as *.com*, *.net*, and *.org*, and the statistics for different gTLDs can be found in Table 5. The *.com* domains account for almost half of the whole MX domain list, so it is not surprising that the *.com* domains have the most domains which support DKIM. If we focus on the adoption rate, *.edu* shows the highest percentage (around 71%) of domain names that have enabled DKIM verification. And also, the percentages of DKIM-enabled domain names under *.com* and *.org* are both above the average adoption rate (37.0%), which are 38.6% and 41.4% respectively.

We also analyze the DKIM deployment of domains from the country level. We aggregate all domain names by country-code TLDs (ccTLDs) and find that DKIM adoption rates vary from the domains of different countries (Table 6). The *.ru* domains account for the most domains that support DKIM

Table 5: DKIM Adoption Rate among Multiple gTLDs.

gTLD	MX Domains	w/ DKIM (%)
.com	371,040	143,156 (38.6%)
.org	33,271	13,787 (41.4%)
.net	33,101	9,926 (30.0%)
.info	5,531	1,443 (26.1%)
.co	3,559	1,453 (40.8%)
.edu	3,062	2,183 (71.3%)
.biz	1,955	534 (27.3%)
.gov	810	431 (53.1%)

Table 6: DKIM Adoption Rate among Multiple ccTLDs.

ccTLD	Country	MX Domains	w/ DKIM (%)
.ru	Russia	34,754	12,107 (34.8%)
.de	Germany	25,105	5,744 (22.9%)
.jp	Japan	17,740	2,467 (13.9%)
.uk	United Kingdom	15,496	7,058 (45.6%)
.br	Brazil	13,990	6,737 (48.2%)
.fr	France	11,012	4,141 (37.6%)
.au	Australia	7,452	4,363 (58.6%)
.cn	China	5,439	422 (7.8%)

under all ccTLDs, followed by *.uk*. Besides, we find DKIM is very popular in Australia, and 58.6% of their MX domains support DKIM. It is worth mentioning that most domains in China and Japan have not supported DKIM yet, and the DKIM adoption rate among *.cn* domains is only around 8.0%.

5 Grammatical Analysis of DKIM Records

In this section, we perform a grammatical analysis of the DKIM records for Alexa top 1 million domains and find the records of 8,147 (2.9%) domains are misconfigured. Some types of misconfigurations are as follows, and Table 7 shows our analysis results.

Table 7: Numbers of Misconfigured Domains.

Misconfiguration Type	# Domain
Abnormal p Field	3,292[1]
- Missing p Field	619
- An empty p Field	172
- Public Key Errors	2553
Invalid Tags	1,967
Multiple DKIM Records For One Selector	2,522
RSA-SHA256 Unsupported	550
Others	504

[1] A domain may be configured with multiple abnormal p field types.

Invalid Tags. As can be seen from Figure 3, DKIM records should consist of different elements in the form of `tag=value` pairs. When parsing DKIM records, we find that there are invalid tags in the records of 1,967 domains. These invalid tags contain characters that are not compliant with RFC regulations, such as `",\,<,;`, which may affect our correct extraction of tags.

Abnormal p Field. In a DKIM record, `p` field represents the public key that recipients use to verify the DKIM signatures. RFC 6376 stipulates that `p` field must appear in DKIM records [13]. If the public key data is abnormal, recipients can not verify the DKIM signatures correctly.

However, we find that 3,292 domains, among Alexa top 1 million domains, use abnormal public keys in their DKIM records that can not be parsed correctly. The abnormal cases include (1) missing `p` field, (2) empty `p` field and (3) public key errors. Missing `p` field means there is no `p` field in the DKIM records. Public key errors mean the keys are in invalid formats, which include public keys with incorrect padding, misuse of quotation marks or escape characters, and some obviously wrong configurations like `p=none`. An empty `p` field may not cause security issues because RFC 6376 [13] stipulates that an empty value means this public key has been revoked, while the other two misconfigurations involving 3,172 domains are obviously wrong, which can make DKIM signature invalid.

Multiple DKIM Records For One Selector. RFC 6376 stipulates that DKIM records must be unique for a particular selector explicitly [13]. However, it does not provide a specific description of which record should be selected if more than one exists. We find 2,522 domain names (within Alexa Top 1M) that deploy multiple DKIM records using one selector in the real-word. For example, *m1* is a selector used by *microsoft.com*, but there are two DKIM records in the TXT records of *m1._domainkey.microsoft.com*. In this situation, the DKIM validation result is determined by specific implementations, since it is undefined in RFC standards.

According to our experiments, the implementation varies from email services. Google and Netease (163.com) will pass the verification of DKIM signatures, only if the last DKIM record is correct. Meanwhile, mail.ru will regard such a signature as valid as long as one of the multiple DKIM records is valid. Outlook and Yahoo! will consider a signature with multiple DKIM records invalid.

RSA-SHA256 Unsupported There are two algorithms to generate DKIM signatures, defined in RFC 6376 [13], including RSA-SHA1 and RSA-SHA256. It is strongly encouraged that signers should use RSA-SHA256, because it is proved that SHA-1 is not as collision-resistant as expected [27], and the suggested algorithms in DKIM have been updated in the most recent RFC 8301 [20] In January 2018. RFC 8301 stipulates that signers *must* use rsa-sha256, while rsa-sha1 *must not* be used for signing or verifying. However, we find that 550 domains only support the RSA-SHA1 al-

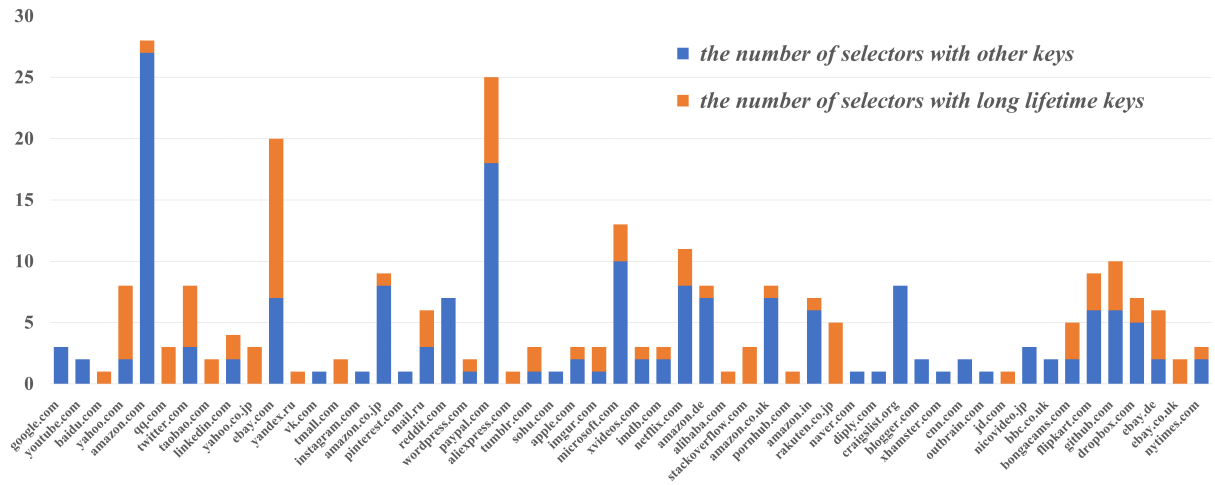

Figure 6: Selector Numbers for Popular Domains. We analyze DKIM keys lifetime of Alexa top 100 domains. The red bars represent the number of selectors, which have not changed their DKIM keys for a long time (5 years here).

gorithm among Alexa top 1M domains, of which the email service administrator should update the algorithm as soon as possible. We will discuss the risk of using the outdated hash algorithm in Section 7.

There are also some other types of misconfigurations in DKIM records and some of them are even confusing. For example, we notice that some records mix DKIM records with SPF or DMARC records, such as `v=spf1; k=rsa; p=MIGFMA...`. Besides, we find the `v` field value of a few records is `DKIM2`, which has not been proposed yet. These issues show that some email service administrators configure DKIM records carelessly, which can cause DKIM signatures invalid.

6 DKIM Key Management Issues

In this section, we focus on how administrators manage DKIM keys, mainly based on passive DNS sources that cover 3,627,871 FQDNs in total. We found three types of key management issues: 1) long lifetime keys; 2) shared keys; 3) weak keys.

6.1 Long Lifetime Keys

DKIM keys should be rotated on a routine basis to balance the security risk of compromised keys and operational effort [3]. We measured the lifetime of DKIM keys by examining their occurrence period in our passive DNS datasets. If a domain adopts multiple DKIM selectors, we will regard the longest selector lifetime as its DKIM key lifetime. The results are shown in Table 8 by year, and we see that 312,852 (8.6%) domains deploy at least one DKIM key whose lifetime is over five years.

Especially, using long-lifetime keys is common even for the most high-profile domains. Figure 6 shows the number of long lifetime keys of 54 domains within Alexa top 100 covered by passive DNS data, including well-known apex domains like baidu.com, yahoo.com, and amazon.com. We find that 10 out of Alexa top 20 domains have not rotated their keys in the past five years, while the percentage is 68.5% out of the above 54 domains (within Alexa Top 100).

For a more accurate evaluation, we re-measure the DKIM keys of all domains using the popular selectors (mentioned in Section 3.2) on Feb 4, 2021. Since we can not get the initial DKIM configuration time of all domain names, neither through passive analysis nor active scanning, our measurement results only show the lower bound of the key's lifetime. Thus, the real lifetime of DKIM keys can be longer than our results.

Table 8: DKIM Key Lifetime in Passive DNS Dataset

DKIM Key Lifetime(year)	# Domain	%
≥ 1	793,679	21.9%
≥ 2	652,742	18.0%
≥ 3	521,033	14.4%
≥ 4	414,022	11.4%
≥ 5	312,852	8.6%

[1] The number of domains with long lifetime DKIM keys is a subset of those with short ones. For example, lifetime ≥ 2 is a subset of lifetime ≥ 1.

Root Causes of Using Long Lifetime Keys. Delving into the root causes, we find no update mechanism for DKIM keys like the policies designed for the public key infrastructure (PKI). RFC 6376 recommends using new selectors to replace public

keys regularly. When administrators update DKIM keys, the old selectors should be held on for a *transition period* to make sure that the emails with old DKIM signatures can be verified.

However, it will involve two potential problems if the updating mechanism for DKIM keys is unclear:

First, historical public keys are not revoked because of the unclear transition period. Even if the DKIM keys are changed voluntarily, administrators may not revoke historical public keys because they can not decide a proper time for the *transition period*. Thus, the historical keys still exist and can be accessed. It is maybe one of the reasons why the majority of DKIM keys have a very long lifetime. For example, a selector for google.com called "20120113", named by its creation date, has hardly been queried since January 2017, based on our Passive DNS dataset. Meanwhile, the number of requests for another selector called "20161025" began increasing dramatically. The request frequency for these two selectors is shown in Figure 7. We find two queries for selector "20120113" in July 2018, indicating the DKIM records still existed at that time even if it has almost no longer been used since January 2017. Besides, a test selector for microsoft.com called "testarcselector01" only appeared on Mar 28th, 2019, which can still be used to verify the DKIM signatures from microsoft.com, with the same authority as other selectors. From the selector name, we infer that the record was used to test for the ARC protocol [8] and administrators may forget to delete it.

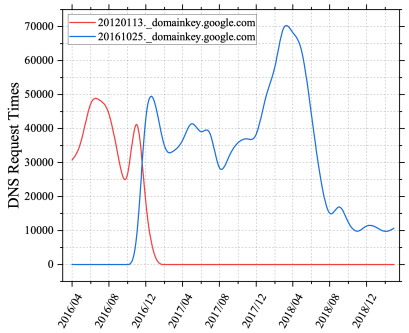

Figure 7: Lifetime of Two DKIM Selectors Used by Google. We analyze the DNS request times of Google's two DKIM keys from Apr. 2016, to Mar. 2019, by month, based on the PassvieDNS dataset.

Second, some domains are configured with multiple selectors. In our datasets, we find 142,073 domain names have more than one selector. For example, eBay has at least 20 selectors, of which 13 selectors have not changed their DKIM keys in the past five years. Figure 8 illustrates the distribution of domains with multiple selectors, and we find 4,195 domains with more than five selectors and 1,333 with even more than ten selectors. There are two potential reasons for these domains to deploy multiple selectors: (1) RFC 6376 [13] recommends DKIM signature signers should not set old selectors for new DKIM keys. Otherwise, the emails with old DKIM signatures can no longer be verified so that recipients can not distinguish spoofing emails from those signed with historical keys. (2) Distributed organizations tend to choose different selectors and key pairs among regions or email servers. However, configuring a domain with too many selectors can increase the security risks since the leakage of any one private key will help attackers craft spoofing emails that can pass the DKIM verification.

Case Study. We discovered a case to demonstrate the security risk of the long lifetime keys. Previous work [11] found Zoho.com was vulnerable to DKIM signature spoofing attacks because of the "l=" tag. We revisited the problem two years later and found that although Zoho.com removed the "l=" tag and used a new selector to sign its outgoing email, it forgot to remove the old DKIM public key from DNS records, which means old emails with "l=" tag can still pass DKIM validation. Attackers could exploit this to replay old emails to bypass DMARC and spoof DKIM signatures. We have reported this bug to Zoho.com, who have fixed it and rewarded us $200 for the report.

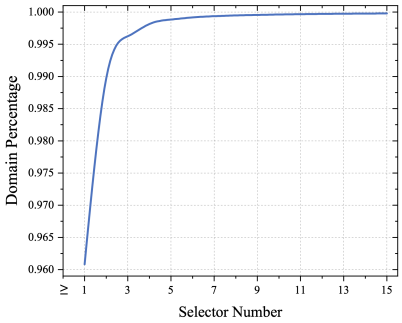

Figure 8: Cumulative Distribution of Domain Names with Multiple DKIM Selectors.

6.2 Shared Keys

Prior works have demonstrated that sharing keys may introduce security issues in the PKI ecosystem [10, 12]. However, there seems to be no in-depth understanding about the prevalence of shared keys in the DKIM deployment. As such, we conduct a measurement study to evaluate key sharing issues, and the results show that the shared keys issues is prevalent in DKIM ecosystem.

We extract DKIM records from passive DNS sources and group domains by their DKIM keys respectively. Notice that we regard domains using the same public key as sharing DKIM keys. In total, we find 61,062 DKIM keys shared by more than one domain and 2,427,682 (66.9%) domains using shared DKIM keys in our datasets. Table 9 shows the top 10 domain groups that share DKIM keys. Especially, there is a

DKIM key shared by over 289,120 domains, which is used for Google's email service.

Table 9: Top 10 Domain Groups Using Shared DKIM Keys.

Group	# Domain	Email Service
1	289,120	Google
2	11,130	Strato
3	8,741	Mailchimp
4	6,979	Sendgrid
5	4,965	Pardot.com
6	4,408	Strato
7	4,156	Sendinblue
8	4,149	Exacttarget
9	3,893	Chinamail
10	1,906	Emarsys

To understand key sharing in more detail, we further investigate domain groups using the shared keys. There are two main situations:

First, multinational companies or organizations allocate the same DKIM keys to their controlled domains for management convenience. For example, PayPal uses the same DKIM key for its domains in different regions, such as `paypal.com.cn` in China and `paypal.com.sg` in Singapore.

Second, vendors of email services (e.g., Gmail, Chinamail) and email marketing services (e.g., Sendgrid, Salesforce) may provide users with DKIM signature services to increase the probability of emails they send reaching the inbox. In this situation, clients of these email services are generally configured with the same selector and DKIM keys.

Case Study. In our passive DNS dataset, there are 4,965 domains configured with the same public key and the same selector, such as falkonry.com, promodel.com, and polarislab.com. Leveraging Google search results, we find they are belongs to the same email marketing service, named Pardot. What is worse, the DKIM public key they adopting is 1024 bits and has probably not been changed for ten years.

6.3 Weak Keys

We analyze the DKIM key length of domains in passive DNS sources. Here, if a domain is configured with multiple selectors during the same period, we regard the shortest DKIM key as the key length of the domain because real attackers always target the weakest points to launch attacks.

Our research shows 84% of 3,627,871 domains still use DKIM keys that are less or equal to 1024 bits, while 5,399 domains use the DKIM keys that are even less than 512 bits. However, weak keys are not encouraged for the current DKIM practices. National Institute of Standards and Technology (NIST) has recommended against using 1024-bit keys since December 31, 2013. RFC 8301 also points out that short RSA keys more easily succumb to offline attacks, and signers *should* use RSA keys of at least 2048 bits [20].

Table 10: DKIM Key Length in PassiveDNS.

DKIM Key Length	# Domain	%
$len = 2048$	579,032	16.0%
$1024 < len < 2048$	6,611	0.2%
$len = 1024$	3,006,398	82.9%
$512 < len < 1024$	30,431	0.8%
$len \leq 512$	5,399	0.2%

To know whether email service administrators have implemented the best DKIM practices, we further investigate key lengths of newly added DKIM keys for each year. We regard the earliest timestamp of a DKIM record as the DKIM key configuration time. Figure 9 shows the number and the percentage of different DKIM key lengths for every year. To better present the changes of different key lengths in the figure, we used log_2^{number} when calculating the percentages since key numbers with different lengths differ by orders of magnitude. We can see that 1024-bits DKIM keys are still the mainstream of in current practices. In addition, DKIM keys are becoming more and more secure, as we notice the proportion of keys no longer than 512 bits has decreased, and that of 2048 bits has increased. The results demonstrate that administrators tend to use more secure keys when updating DKIM configurations for email security. Considering the improvement of computing power, we suggest this updating process should be accelerated.

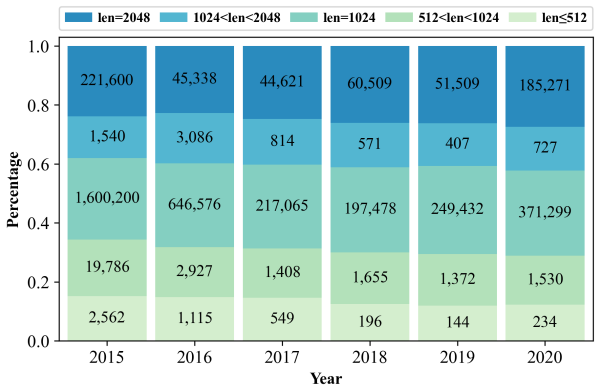

Figure 9: Number and Percentage of Different DKIM Key Lengths from 2015 to 2020. The percentage is calculated by the log_2^{number} of different DKIM key lengths.

7 DKIM Signature Issues

DKIM provides many user-defined options, so it is prone to some configuration issues. We can analyze issues of DKIM

signatures in practice based on 460 million DKIM signatures derived from real emails provided by Coremail. We find 94.2% domains have DKIM signature issues, including weak DKIM signatures, insecure "l=" tags, and outdated hash algorithms.

7.1 Weak DKIM Signatures

DKIM signatures should sign important email headers to protect the content integrity of emails and avoid being abused for replay attacks. However, RFC 6376 [13] only specifies the `From` field to be *must* signed. Although it also recommends that 20 headers *should* be signed in the DKIM signatures, we find no domain in the Coremail DKIM data signing all of the 20 headers and the header fields signed in DKIM signatures vary among different email services.

However, it is insecure if some important fields are not signed, since attackers can arbitrarily tamper with the value of these fields to construct spoofing emails they need. For example, an attacker can replace the reply address of a legitimate email with their address if the `Reply-To` field is not signed, and thus, any reply email will be sent to the attacker.

We investigate all of the 20 headers recommended to be signed in the RFC 6376 [13] and divide them into two categories, as shown in Table 11. The meaning of these headers are introduced in Appendix B. We suggest that DKIM signatures should at least sign the fields in the first class, because they represent the parts that users can easily notice from user interfaces. If they are not signed, a DKIM signature can be abused for replay attacks and the user will likely notice that a spoofing email has passed DKIM verification. The technical details of these attacks are discussed in Chen's paper [11]. Besides, as aforementioned, the `Reply-To` field should also be signed to avoid replying to an attacker's address. Thus, we regard the signatures that do not sign these seven fields as weak DKIM signatures. On this point, we also had a discussion with Coremail's engineers and they agreed with our classification. Due to the complexity of email headers, a systematic security analysis of all email headers is necessary. The classification results should also be dynamically maintained.

Table 11: Headers Recommended to be Signed in RFC 6376.

Class	Field Name
1	From, Reply-To, Subject, Date, To, Cc, Content-Type
2	Resent-Date, Resent-From, Resent-To, Resent-Cc, In-Reply-To, References, List-Id, List-Help, List-Unsubscribe, List-Subscribe, List-Post, List-Owner, List-Archive

In addition, we also parse the email headers from "h=" tags in DKIM signatures from Coremail. We calculate the percentage of each header and show the results in Table 12. We find that 2,074,178 (94.1%) domains use weak DKIM signatures. Almost all the domains have signed `From` and `Subject` fields, while only 29 domains (e.g., mikegmarketing.com and nimbios.org) have not. However, there are still domains that have not signed `Cc` fields (91.5%), `Reply-To` fields (88.6%), `Content-Type` fields (32.5%), `Date` fields (24.2%), and `To` fields (13.3%).

Table 12: Top 10 Email Headers in DKIM Signatures.

Rank	Field Name	%
1	From	100.0%
2	Subject	99.7%
3	To	86.7%
4	Date	75.8%
5	Mime-Version	73.6%
6	Message-Id	73.3%
7	Content-Type	67.5%
8	Content-Transfer-Encoding	19.5%
9	X-Ms-Exchange-Senderadcheck	12.5%
10	Reply-To	11.4%

Oversigning. RFC 5322 explicitly specifies that there should be only one `From` header in an email [29]. However, email services are liberal in what they receive. Previous work shows that emails with multiple `From` headers can still be accepted by most popular email services [31], so attackers can exploit this feature to conduct email spoofing attacks [1, 11, 31]. To prevent adding duplicate header fields, DKIM designers proposed a method of oversigning [13], which means a header name should appear in "h=" tags once more than the actual number of that header in an email. For example, suppose that an email contains `From`, `To` and `Subject` headers, and each header appears only once. If email services use the oversigning mechanism to protect `From` and `To` fields, these two headers should be listed twice in the "h=" tag, such as "`h=from:from:to:to:subject:...`". When the oversigning mechanism is supported, the signer will calculate DKIM signatures regarding the extra headers as empty fields. In this way, attackers can not add extra headers to conduct email spoofing attacks.

We analyze how many domains use oversigning to protect themselves against multiple headers attacks. RFC 5322 (section 3.6) specifies there are 11 headers should occur at most one time in email headers. Therefore, we regard these headers as being oversigned if these headers appear more than once in "h=" tags of DKIM signatures. We found only 47549 (2.2%) domains are found to be protected, as shown in Table 13. For the seven essential headers we suggest to be signed, we find 47,334 domains use oversigning to protect the `From` field, while only 603 domains protect the `Reply-To` field. Fortunately, some well-known service providers have started to use this mechanism to protect their outgoing emails, such as yahoo.com and mail.ru. The emails sent by these

providers account for 12.8% in our DKIM signature datasets.

Table 13: Top 10 Headers Protected by Oversigning Mechanism.

Rank	Field Name	# Domain	%
1	From	47,334	99.5%
2	Subject	16,597	34.9%
3	Date	11,144	23.4%
4	To	5,913	12.4%
5	Message-Id	5,068	10.7%
6	In-Reply-To	2,611	5.5%
7	References	2,487	5.2%
8	Cc	2,004	4.2%
9	Reply-To	603	1.3%
10	Sender	165	0.3%

7.2 Insecure "l=" Tags

In DKIM records, "l=" tags are used to limit the length of email bodies that should be calculated for DKIM signatures explicitly. If the body length is not specified, the entire message body will be signed. This tag is designed for increasing DKIM signature robustness since the unsubscribe information can be added to the end of email messages, and some antivirus products may also add the notification like "the email was scanned by product ABC". However, using "l=" tags are proved to be insecure, since the misuse of the "l=" tag can allow displaying fraudulent content to end-users without breaking the DKIM signature, as discussed in Section 8.2 in RFC 6376 [13]. Besides, attackers can craft a spoofing email exploiting the insecure "l=" tag and multiple `Content-Type` headers [11], without breaking the original DKIM signature. Our results show that 6,860 (0.3%) domains still use "l=" tags in DKIM signatures, of which 1,273 domains are within Alexa top 1 million domains.

7.3 Outdated Hash Algorithms

Hash algorithms are essential to creating digital signatures, the collision resistance of which can directly affect the protective effect of digital signature algorithms. SHA-1 and SHA-256 are commonly used in DKIM, however, SHA-1 is not as collision-resistant as expected, on which theoretical attacks have been known since 2005 [33]. In recent years, a series of works [22, 23, 32] have shown that attacking on SHA-1 is becoming practical. Besides, SHA-1 was officially deprecated by NIST in 2011 [6]. RFC 8301 [20] has also recommended that rsa-sha1 must not be used for signing or verifying DKIM signatures in 2018. However, the rsa-sha1 algorithm is still widely used. We find 1,451,956 (65.9%) domains still use rsa-sha1 to generate DKIM signatures, of which 3,292 domains are within Alexa top 1M domains.

Theoretically, adopting a weak hash algorithm for DKIM signatures may lead to attacks enabled by hash collisions [22, 23, 32]. If an adversary can find a meaningful hash collision of the given email's body, he/she can replace email bodies and craft spoofing emails without breaking DKIM signatures, which will seriously break the protection of DKIM. Though it is challenging for practical attacks, we suggest updating the relevant algorithms in practices before any actual attack is found.

8 Discussion

8.1 Limitations

Data Collection. We have to admit our DKIM records and signatures can be biased due to the geo-location of the affiliated DNS resolvers and email servers. However, our dataset still shows representative results from three aspects: (1) Because this study needs DKIM data from diverse domains, the large scale and the long period for passive DNS data collection can meet the requirement. (2) Coremail occupies a considerable share of the Chinese email market, its clients can receive emails from all over the world. Thus, the Coremail DKIM signature dataset is an effective supplement to the passive DNS data. For example, the passive DNS data can only cover the 8.7% of Alexa top 1M domains, while the number is up to 10.2% when adding the Coremail data. (3) We also conduct an active scanning on Alexa top domains to eliminate the dataset limitation with the best effort.

Above all, we have tried our best to measure the DKIM deployment among Alexa top 1 million domains. We admit that our measurement result is only the lower bound of the real-world DKIM adoption, while it is enough for further analysis.

Canonicalization. Canonicalization algorithm is an important component of the DKIM protocol, which makes it possible to verify DKIM signatures, even if the SMTP gateway has slightly changed the source code of the email. Due to the consideration of privacy issues, we only get and use the DKIM-Signature headers provided by Coremail and cannot access other email headers and email bodies. As a result, our study may not evaluate the influence of various canonicalization algorithms, which is another limitation of our study. We conducted a preliminary evaluation of the usage of different canonicalization algorithms. The results are described in the Appendix A.

8.2 Mitigation

Disclosure. We have tried to responsibly report all vulnerabilities we found to the relevant email administrators. It is non-trivial since the total number of involved domain

names is extremely large, and we lack some contact information of these email administrators. Thus, we first report the vulnerabilities of DKIM deployment to some reputable email service vendors, such as Gmail and Sendgrid. Second, for other involved domains, we collect email addresses via the Whois API and contact them with detailed vulnerability reports. So far, we have received responses from Gmail, Mailchimp, Sendgrid and Salesforce. They acknowledged our report and actively discussed the potential impact of these problems with us. Besides, we also received feedbacks from 24 relevant email administrators, including beyovantage.com, secureworks.com, pax.com, hbtc.com. They acknowledged our report and particularly thank us for reporting these vulnerabilities to them.

Online Detection Tool. Reasons for the unsatisfactory deployment of DKIM include (1) the differences in administrators' understanding of DKIM deployment, (2) the lack of unified deployment recommendations, and (3) no easy-to-use detection tool for administrators to validate DKIM deployment. As such, we develop an online tool for DKIM deployment based on our measurement and analysis scripts, which can be accessed at *https://nospoofing.cn*. This tool can help email administrators check and deploy their DKIM records and further improve the status quo of DKIM deployment.

Our online detection tool provides two query methods. Users can provide the tool with a domain name and its corresponding selector or send an email to our designated email address to query the current DKIM deployment situation of this domain. Our tool can do the grammar check and analyze the key strength and judge whether the DKIM signatures have the security issues mentioned in this paper. Compared with some existing tools, [3] our tool can not only test whether a domain has deployed DKIM, but also conduct a comprehensive security analysis of DKIM records and DKIM signatures, and give corresponding deployment recommendations.

8.3 Recommendations

We find that some implementation-level problems can be avoided by modifying the protocol, so we propose two improvements: (1) adding an expired date for DKIM keys and (2) setting "oversigning" as the default mechanism. These suggestions only need incremental changes, which can be compatible with old versions of the protocol. This iterative process can be achieved smoothly by updating the relevant verification library.

DKIM Key Expiration Date. It is because RFC 6376 has not specified a clear *transition period* for old keys that long lifetime DKIM keys are common in practice. Besides, in the DKIM ecosystem, a considerable number of domains will not remove their DKIM keys from DNS servers in time, even though the keys are revoked. For example, in Figure 7, though Google has started to use a new key, the historical one has still been kept for one and a half years after being abandoned. Thus, a feasible solution is to add a field of DKIM key expiration date to DKIM records, which can help alleviate the problem of the unclear transition period and promote regular key replacement. The sending services can decide whether to use this field. If this field is used, email services should sign the `Date` field.

Email senders should stop using a public key to generate DKIM signatures two months before the expiration time to ensure that the historical DKIM signature can be verified. In the PKI ecosystem, the famous certificate authority *Let's Encrypt* suggests its users update their certifications one month before they expire. Moreover, our industrial partner, Coremail, has confirmed that it is relatively reasonable to set the transition period to two months based on their deployment practices. Email recipients should first determine whether the DKIM signatures of currently received emails have expired based on this field when verifying DKIM signatures.

We tested 11 well-known mail services (e.g., Gmail, Yahoo, and Yandex) and open-source DKIM libraries (OpenDKIM, DKIMproxy) for backward compatibility. We found that all of them are compatible with this solution. In detail, we set up an email service, added an `expired-date` field to our DKIM record, and sent emails signed with our DKIM key to famous email services. The email transfer and the DKIM verification worked well together with this solution.

Default Oversigning Mechanism. The oversigning mechanism is helpful to protect users from the email spoofing attacks that use multiple email headers [11, 31]. However, our measurement results show that few email administrators are aware of this kind of email spoofing attack.Thus, it is better to change the implementation of popular DKIM libraries. The DKIM libraries can set the oversigning mechanism as default when signing DKIM signatures. That is to say, signers should use the default oversigning mechanism to protect essential headers, including `From`, `To`, `Subject`, `Content-Type`, `Reply-To`, `Date`, and `Cc`. The only modification signers need to do is list the headers once more in the `h` field than it should be.

RFC 5322 [29] specifies that `From`, `To`, `Subject`, `Cc`, and `Reply-To` should occur once at most. Thus, for general cases, it is enough for signers to list these headers twice in the `h` field. Email service administrators can decide whether to use the default oversigning mechanism to protect other headers, and they should not use it to protect the header they want to add or change. It is only a small change for signers and will not affect the DKIM verification process, so we consider it backward compatible. Besides, this change will significantly improve the protective effect of DKIM signatures and prevent DKIM signatures from being used for replay attacks.

[3] https://poste.io/dkim, https://www.mail-tester.com, and https://internet.nl

9 Related Work

DKIM Deployment To the best of our knowledge, this paper is the first large-scale, longitudinal analysis of DKIM deployment and related misconfiguration issues. SMTP extensions such as DKIM, SPF and DMARC are used to provide security properties for email transport. There are a few measurement studies on the deployment of SPF, DKIM, and DMARC [14, 15, 18]. Their results indicate that the adoption and enforcement of these extensions need improvement. However, they pay more attention to the deployment rate of the email security protocol, rather than the misconfiguration issues. In addition, the previous work has less analysis of DKIM deployment, due to the difficulty to obtain DKIM data through active scanning.

Among them, the most similar work is that Durumeric et al. [14] published studies on the measurement of email security protocols (SPF, DKIM and DMARC) in 2015. They analyzed the security configurations of top email providers based on SMTP connections from and to the Google email server between January 2014 to April 2015. After 5 years, a new measurement is needed at this time to analyze the current deployment of DKIM. Compared with their work, our dataset has a larger amount data and a long time span. Our dataset contains PassiveDNS data over 5 years and 460 million DKIM signatures derived from email system in practice.

Email Security Email has long been fraught with security issues such as email spoofing attacks [11, 18, 31]. To address these problems, various security extensions have been proposed and standardized. At present, SPF, DKIM and DMARC protocols are the most widely used ones. Among them, DKIM is an effective way to authenticate a sender and verify the integrity of received emails.

Recently, some studies focus on the email spoofing attacks in practice. Hu et al. [18] analyzed how email vendors detect and handle spoofing emails through an end-to-end email spoofing experiment. Shen et al. [31] presented a series of new attacks that can bypass SPF, DKIM, DMARC and user-interface protections through a systematic analysis of the email delivery process. They conducted a large-scale analysis of 30 popular email services and 23 email clients, and found that all of them are vulnerable to certain types of attacks. Chen et al. [11] introduced the ambiguous-replay attacks with seemingly valid DKIM signatures from legitimate domains. Jens et al. analyzed the security issues in the OpenPGP and S/MIME protocols. They devised a series of practical forgery attacks against various implementations of OpenPGP and S/MIME and proposed countermeasures [25, 26, 30]. Unlike prior works, our work shows the current deployment status of DKIM in practice. We reveal many DKIM misconfiguration issues in the real email ecosystem. These results highlight systemic problems, which motivate improved automation and auditing of DKIM management.

10 Conclusion

In this paper, we illustrate the DKIM deployment and its potential security issues. We perform a large-scale measurement study on DKIM and show that 28.1% of Alexa Top 1 million domains have enabled DKIM. However, the mismanagement of DKIM is still prevalent in the email ecosystem, including grammar errors, DKIM key management issues, and DKIM signatures issues. The mismanagement weakens the protection of DKIM and poses security risks to email users. We report the vulnerabilities to the email administrators and provide an online tool to ananlyze the deployment of DKIM. We believe this work will improve the deployment of DKIM and inspire the community to work towards securing DKIM deployment.

Acknowlegments

We sincerely thank our shepherd Jörg Schwenk and all the anonymous reviewers for their valuable reviews and comments to improve this paper. We also thank Chenrui Li, Zhuo Li, and Xiang Li for their peer-reviewing and assisting in editing this paper. We are grateful for the support from Qi-Anxin, 360 and Coremail in data collection.

This work was supported in part by the National Natural Science Foundation of China (U1836213, U19B2034, 62102218). Baojun Liu was partially supported by the Shuimu Tsinghua Scholar Program. Any opinions, findings, conclusions or recommendations expressed in this paper do not necessarily reflect the views of the NSFC.

References

[1] Breaking dkim - on purpose and by chance. https://maulwuff.de/research/breaking-dkim-on-purpose-and-by-chance.html. Accessed: November 28, 2020.

[2] Business e-mail compromise the 12 billion dollar scam. https://www.ic3.gov/Media/Y2018/PSA180712. Accessed: Jan 28, 2021.

[3] Dkim key rotation best common practices. http://www.m3aawg.org/DKIMKeyRotation. Accessed: December 18, 2021.

[4] Passive dns historical internet database. https://www.farsightsecurity.com/solutions/dnsdb/. Accessed: November 28, 2020.

[5] Research: Crisis of fake email continues to plague industries worldwide. https://www.valimail.com/press/research-crisis-of-fake-email-continues-to-plague-industries-worldwide-2/. Accessed: Jan 28, 2021.

[6] Research results on sha-1 collisions. https://csrc.nist.gov/news/2017/research-results-on-sha-1-collisions. Accessed: Oct 11, 2021.

[7] Spam and phishing in q1 2019. https://securelist.com/spam-and-phishing-in-q1-2019/90795/. Accessed: Jan 28, 2021.

[8] Kurt Andersen, Brandon Long, Seth Blank, and Murray S. Kucherawy. The authenticated received chain (ARC) protocol. *RFC*, 8617:1–35, 2019.

[9] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. EXPOSURE: finding malicious domains using passive DNS analysis. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, 6th February - 9th February 2011*. The Internet Society, 2011.

[10] Frank Cangialosi, Taejoong Chung, David R. Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. Measurement and analysis of private key sharing in the HTTPS ecosystem. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 628–640. ACM, 2016.

[11] Jianjun Chen, Vern Paxson, and Jian Jiang. Composition kills: A case study of email sender authentication. In Srdjan Capkun and Franziska Roesner, editors, *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, pages 2183–2199. USENIX Association, 2020.

[12] Taejoong Chung, Roland van Rijswijk-Deij, Balakrishnan Chandrasekaran, David R. Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. A longitudinal, end-to-end view of the DNSSEC ecosystem. In Engin Kirda and Thomas Ristenpart, editors, *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*, pages 1307–1322. USENIX Association, 2017.

[13] Dave Crocker, Tony Hansen, and Murray S. Kucherawy. Domainkeys identified mail (DKIM) signatures. *RFC*, 6376:1–76, 2011.

[14] Zakir Durumeric, David Adrian, Ariana Mirian, James Kasten, Elie Bursztein, Nicolas Lidzborski, Kurt Thomas, Vijay Eranti, Michael Bailey, and J. Alex Halderman. Neither snow nor rain nor MITM...: an empirical analysis of email delivery security. In Kenjiro Cho, Kensuke Fukuda, Vivek S. Pai, and Neil Spring, editors, *Proceedings of the 2015 ACM Internet Measurement Conference, IMC 2015, Tokyo, Japan, October 28-30, 2015*, pages 27–39. ACM, 2015.

[15] Ian D. Foster, Jon Larson, Max Masich, Alex C. Snoeren, Stefan Savage, and Kirill Levchenko. Security by any other name: On the effectiveness of provider based email security. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 450–464. ACM, 2015.

[16] Hongyu Gao, Vinod Yegneswaran, Yan Chen, Phillip A. Porras, Shalini Ghosh, Jian Jiang, and Hai-Xin Duan. An empirical reexamination of global DNS behavior. In Dah Ming Chiu, Jia Wang, Paul Barford, and Srinivasan Seshan, editors, *ACM SIGCOMM 2013 Conference, SIGCOMM 2013, Hong Kong, August 12-16, 2013*, pages 267–278. ACM, 2013.

[17] Amir Herzberg. Dns-based email sender authentication mechanisms: A critical review. *Comput. Secur.*, 28(8):731–742, 2009.

[18] Hang Hu and Gang Wang. End-to-end measurements of email spoofing attacks. In William Enck and Adrienne Porter Felt, editors, *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, pages 1095–1112. USENIX Association, 2018.

[19] Scott Kitterman. Sender policy framework (SPF) for authorizing use of domains in email, version 1. *RFC*, 7208:1–64, 2014.

[20] Scott Kitterman. Cryptographic algorithm and key usage update to domainkeys identified mail (DKIM). *RFC*, 8301:1–5, 2018.

[21] Murray S. Kucherawy and Elizabeth D. Zwicky. Domain-based message authentication, reporting, and conformance (DMARC). *RFC*, 7489:1–73, 2015.

[22] Gaëtan Leurent and Thomas Peyrin. From collisions to chosen-prefix collisions application to full SHA-1. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 527–555. Springer, 2019.

[23] Gaëtan Leurent and Thomas Peyrin. SHA-1 is a shambles: First chosen-prefix collision on SHA-1 and application to the PGP web of trust. In Srdjan Capkun and Franziska Roesner, editors, *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, pages 1839–1856. USENIX Association, 2020.

[24] John R. Levine. A new cryptographic signature method for domainkeys identified mail (DKIM). *RFC*, 8463:1–7, 2018.

[25] Jens Müller, Marcus Brinkmann, Damian Poddebniak, Hanno Böck, Sebastian Schinzel, Juraj Somorovsky, and Jörg Schwenk. "johnny, you are fired!" - spoofing openpgp and S/MIME signatures in emails. In Nadia Heninger and Patrick Traynor, editors, *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, pages 1011–1028. USENIX Association, 2019.

[26] Damian Poddebniak, Christian Dresen, Jens Müller, Fabian Ising, Sebastian Schinzel, Simon Friedberger, Juraj Somorovsky, and Jörg Schwenk. Efail: Breaking S/MIME and openpgp email encryption using exfiltration channels. In William Enck and Adrienne Porter Felt, editors, *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, pages 549–566. USENIX Association, 2018.

[27] Tim Polk, Lily Chen, Sean Turner, and Paul E. Hoffman. Security considerations for the SHA-0 and SHA-1 message-digest algorithms. *RFC*, 6194:1–7, 2011.

[28] Jonathan Postel. Rfc0821: Simple mail transfer protocol, 1982.

[29] Peter W. Resnick. Internet message format. *RFC*, 5322:1–57, 2008.

[30] Jörg Schwenk, Marcus Brinkmann, Damian Poddebniak, Jens Müller, Juraj Somorovsky, and Sebastian Schinzel. Mitigation of attacks on email end-to-end encryption. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, pages 1647–1664. ACM, 2020.

[31] Kaiwen Shen, Chuhan Wang, Minglei Guo, Xiaofeng Zheng, Chaoyi Lu, Baojun Liu, Yuxuan Zhao, Shuang Hao, Haixin Duan, Qingfeng Pan, and Min Yang. Weak links in authentication chains: A large-scale analysis of email sender spoofing attacks. In Michael Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 3201–3217. USENIX Association, 2021.

[32] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full SHA-1. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 570–596. Springer, 2017.

[33] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005.

A Canonicalization Algorithms

We analyzed the usage of different canonicalization algorithms in real-world email communication, based on our DKIM signatures datasets. The result is shown in Table 14. We found that most domains apply a "relaxed" canonicalization algorithm for both email bodies and headers.

Table 14: The Usage of Different Canonicalization Algorithms.

Canonicalization Algorithms	# Domain
relaxed / relaxed	1,620,536
simple / simple	527,429
relaxed / simple	61,471
simple / relaxed	1,110

B Email Headers

We summarize the meaning of each field in Table 15.

Table 15: The meaning of each field in Table 11.

Field Name	Meaning
From	Specifies the author(s) of the message, that is, the mailbox(es) of the person(s) or system(s) responsible for the writing of the message.
Reply-To	Indicates the address(es) to which the author of the message suggests that replies be sent.
Subject	Contains a short string identifying the topic of the message.
Date	Specifies the date and time at which the creator of the message indicated that the message was complete and ready to enter the mail delivery system.
To	Contains the address(es) of the primary recipient(s) of the message.
Cc	Contains the addresses of others who are to receive the message, though the content of the message may not be directed at them.
Content-Type	Specifies the nature of the data in the body of an entity by giving media type and subtype identifiers, and by providing auxiliary information that may be required for certain media types.
Resent-Date	Indicates the date and time at which the resent message is dispatched by the resender of the message.
Resent-From	Contains the mailbox of the individual doing the resending.
Resent-To	Function identically to the “To”, except that it indicates the recipients of the resent message.
Resent-Cc	Function identically to the “Cc”, except that it indicates the recipients of the resent message.
In-Reply-To	Identifies the message (or messages) to which the new message is a reply.
References	Identifies a thread of conversation.
List-Id	Provides an identifier for an e-mail distribution list.
List-Help	Provides an access point to detailed user support information, and accommodate almost all existing list managers command sets.
List-unsubscribe	Describes the command (preferably using mail) to directly unsubscribe the user (removing them from the list).
List-Subscribe	Describes the command (preferably using mail) to directly subscribe the user (request addition to the list).
List-Post	Describes the method for posting to the list.
List-Owner	Identifies the path to contact a human administrator for the list.
List-Archive	Describes how to access archives for the list.