The Fragility of DNS-Based Security Under Imperfect DNS Operation

Tino Hager
Mailtower.app
tino@nager.software

Ronald Petrlic
Nuremberg Institute of Technology
ronald.petrlic@th-nuernberg.de

***Abstract*—The widespread enforcement of email authentication mechanisms such as SPF, DKIM, and DMARC by major email providers has become a cornerstone in the fight against email spoofing. However, since these policies have been rigorously checked in practice, a paradoxical problem has emerged: emails that are correctly authenticated and fully compliant with all policies are nevertheless rejected. In particular, *temp errors* appear to occur arbitrarily and can account for substantial email delivery failures. To date, no systematic explanation for this phenomenon has been provided.**

In this paper, we present the first comprehensive study that shows that these errors are not caused by the authentication mechanisms themselves, but by limitations and failures in the underlying DNS infrastructure. Our measurements reveal that the DNS zones of some—especially large—organizations are overcrowded with TXT records used for domain verification. We show that the resulting number and size of DNS records can directly interfere with SPF evaluation, leading to rejected emails. Furthermore, we identify issues in the DNS infrastructure of Amazon Web Services, where oversized DNS responses can trigger errors and, consequently, render emails undeliverable.

Beyond SPF, we show that DKIM configurations also contribute to delivery failures: RSA key lengths exceeding 2000 bits—despite being considered state of the art—can already result in non-delivery due to excessively large DNS responses. Finally, we are the first to uncover that Microsoft's Exchange Online infrastructure exhibits shortcomings in handling long DNS responses, which explains a significant number of email delivery failures, particularly for large enterprises with extensive DNS configurations.

Overall, our findings provide a new perspective on the reliability of modern email authentication and demonstrate that DNS scalability and implementation limitations represent a critical, yet previously overlooked, root cause of authentication-related email delivery failures.

I. Introduction

The *Domain Name System* (DNS) is a foundational component of the Internet, providing a distributed, hierarchical naming service that maps human-readable domain names to network resources. Designed for scalability and performance rather than strong security guarantees, DNS has nevertheless evolved into a critical infrastructure element that many modern security mechanisms depend upon. Over the past years, an increasing number of security approaches—particularly in the email ecosystem—have begun to rely on DNS as a trusted auxiliary channel or "second factor" for publishing authentication data.

Sender Policy Framework (SPF) uses DNS TXT records to designate which mail servers are authorized to send email on behalf of a domain, allowing receivers to detect forged sender addresses. *DomainKeys Identified Mail* (DKIM) similarly stores public keys in DNS, enabling mail transfer agents to verify cryptographic signatures attached to outgoing messages. *DNS-Based Authentication of Named Entities* (DANE) leverages DNSSEC to bind TLS certificates to domain names, providing authenticated key material that can strengthen or even replace traditional public key infrastructures in email transport security.

While these mechanisms differ in design and purpose, they share a common dependency: the correctness, availability, and integrity of DNS data. As a result, the security of email authentication increasingly hinges on the reliable operation of the DNS itself. This growing reliance underscores the need to closely examine DNS behavior and its impact on the robustness of higher-layer security protocols.

In this paper, we show how DNS problems lead to undeliverability problems in the email ecosystem. While it is a positive development that big email providers like Google, Yahoo, and Microsoft keep strengthening the requirements regarding email authentication via SPF and DKIM, we notice that the practice is not yet ready for a strict enforcement. Email deliverability experts have been noticing that there is an increased number of emails that cannot be delivered to recipients even though all the proposed security mechanisms are in place.

A. Contribution

We are the first to pinpoint the problems of an increasing number of undeliverable *authenticated* emails occurring in practice today. By investigating different email delivery scenarios and looking at DMARC reports in more detail, we are able to pinpoint the problems to the DNS. While the majority of emails is handled by a minority of big email providers worldwide and one might expect that these email providers have a strong technical basis, we show that this is not the case.

Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb) 2026
27 February 2026 , San Diego, CA, USA
ISBN 978-1-970672-06-0
https://dx.doi.org/10.14722/madweb.2026.23041
www.ndss-symposium.org

To the best of our knowledge, we are the first to show that the common practice of putting too much data into the DNS (in the form of TXT entries) results in higher error rates of email authentication with SPF. Moreover, we are the first to find that Microsoft has problems with EDNS in its popular Exchange Online service, reinforcing the authentication problem. Furthermore, we show that state-of-the-art key sizes for DKIM also yield higher error rates in practice, giving administrators reasons to rely on shorter, more insecure key sizes in order not to endanger email deliverability.

B. Paper Outline

The remainder of this paper is structured as follows. In Sect. II, we give a brief overview of state-of-the-art email authentication mechanisms. In Sect. III we present our investigation of the implementation of the mechanisms in practice and the respective results, before presenting possible solutions to the problems in Sect. IV. Related work is presented in Sect. V before we conclude the paper in Sect. VI.

II. Background

The *Domain Name System* (DNS) is a foundational component of the Internet, providing a distributed, hierarchical naming service that maps domain names to IP addresses [1]. Over the past decades, DNS has evolved from a mere name–to–address directory into a versatile metadata distribution and verification channel. Today, DNS serves as a second-factor or out-of-band trust anchor for numerous security mechanisms, including certificate issuance (e.g., ACME DNS-01 challenges) [2], domain authorization, and a wide range of email authentication schemes such as SPF, DKIM, and DMARC [3], [4], [5].

Many of these techniques rely on the *TXT* record type—chosen for its flexibility and universal support across DNS software and hosting providers. As a result, large enterprises commonly accumulate tens or even hundreds of TXT records for different services and verifications [6]. This trend significantly increases the likelihood of oversized DNS responses.

A. DNS Response Sizes, EDNS(0), and DNS Flag Day

Traditional DNS over UDP limited the response sizes to 512 bytes [1]. Beyond this threshold, responses could be truncated, requiring resolvers to retry the query over TCP—an approach that introduces latency and operational overhead. With TXT-heavy domains becoming widespread, large DNS responses became the norm rather than the exception.

To address these limitations, the DNS community introduced *EDNS(0)* in RFC 2671 (later obsoleted and refined in RFC 6891) [7], allowing resolvers and servers to negotiate larger UDP payload sizes (commonly 1232 bytes). EDNS(0) also enabled additional features, such as extended return codes and optional data.

However, inconsistent EDNS support across implementations led to interoperability failures. This motivated several *DNS Flag Day* initiatives, coordinated by major DNS operators, resolver developers, and vendors. DNS Flag Day 2019 and subsequent efforts collectively enforced standards compliance by assuming correct EDNS behavior and no longer accommodating noncompliant servers [8]. This shift improved predictability and performance for large DNS responses—an increasingly important factor as TXT-based security mechanisms proliferate.

B. DNS-Based Email Authentication: SPF, DKIM, and DMARC

1) Sender Policy Framework (SPF): SPF allows domain owners to publish which mail servers are authorized to send messages on their behalf. The policy is stored in a DNS TXT record [3], which specifies mechanisms and modifiers that describe valid sending hosts. During email reception, the receiving Mail Transfer Agent (MTA) queries the domain's SPF record and verifies whether the sender's IP address is authorized. Complex sending infrastructures often lead to complex SPF records with multiple includes, increasing DNS response sizes.

2) DomainKeys Identified Mail (DKIM): DKIM provides cryptographic validation of email authenticity via digital signatures. The sending domain publishes the public key in a TXT record under a selector-specific subdomain (e.g., selector._domainkey.example.com) [4]. Each outgoing email is signed with a private key, and receiving MTAs fetch the public key from the DNS to verify the signature. Longer key sizes, key-rotation strategies, and parallel deployment of multiple selectors often lead to additional DNS overhead.

3) Domain-based Message Authentication, Reporting, and Conformance (DMARC): DMARC is based on SPF and DKIM to define domain-level policies on how receivers should treat failed authentication results [5]. Domain administrators publish DMARC policies in a TXT record at _dmarc.example.com, specifying alignment requirements, reporting endpoints, and desired handling actions (none/quarantine/reject). Like SPF and DKIM, DMARC introduces additional metadata into DNS zones.

4) 2024 Provider Requirements by Google, Yahoo, and Microsoft: In early 2024, major email providers—including Google and Yahoo—announced new policies requiring that bulk senders and many standard senders implement SPF, DKIM, and DMARC to reduce spam, spoofing, and abuse [9], [10]. These requirements included authentication alignment, rate limits, one-click unsubscribe mechanisms, and improved domain hygiene. Domains lacking these DNS-based authentication mechanisms would face message rejections or reduced deliverability.

Microsoft similarly announced its intention to enforce DMARC and related authentication technologies across its email ecosystem, gradually tightening requirements to align with Google and Yahoo's 2024 standards [11]. These coordinated industry moves further reinforce the central role of the DNS as a security assertion channel—and, consequently, the

importance of robust handling of large DNS responses through EDNS(0) and standards-compliant resolver behavior.

III. Investigation

After multiple reports from the email deliverability community (e.g., from different DMARC service providers in Europe and the U.S., the certified senders alliance (CSA), etc.) about an *increasing number of inexplicably rejected authenticated emails*, we started an investigation by examining the data provided by a big DMARC service provider that serves many clients in Europe. We could confirm that there was a percentage of emails—all of which were authenticated with proper SPF, DKIM, and DMARC settings in place—sent by different companies (the clients of the DMARC service provider) to all kinds of receivers, that were rejected by the receiving MTAs for no obvious reason. The DMARC reports, which should indicate the reason for rejection, did not give any indication, though, why these properly authenticated emails from trusted sources have been rejected. Due to the lack of information and the seemingly randomness of rejections, we started to investigate to which extent we could assign the problems to certain common email providers in a first step.

1) The Role of the Email Provider: The analysis of the recipients of authenticated emails that were rejected with a "temp error" revealed that the majority of these recipients employed Microsoft as their email provider. At the same time, we found that Google Mail recipients were not affected to this extent. We decided to use these two email providers as examples for an in-depth investigation to find out the cause for the undeliverable emails. As Microsoft, Google, and Yahoo are the biggest email providers worldwide, it is meaningful to perform an analysis of two of these providers, while being able to transfer the findings to other (smaller) email providers as well.

A. Investigation Environment Setup

Our investigation environment is based on the following setup. We developed a mail server based on the .NET library of *MailKit*[1]. We pursued this approach to have full control over the MTA. In particular, we needed to make sure that the emails sent are consistent. For example, we used the same DKIM signature several times to be able to exclude faults and ensure comparability. The mail server is hosted on a popular German hosting provider. The server is not listed as a spam server and, thus, it can be excluded that emails get rejected for spam reasons, for example. Moreover, the used domains have been in active use by the DMARC service provider (with whom we cooperate in this work) for several years and a huge amount of emails are sent from these domains daily.

We used 6 different domains for our investigation, where the domains' authoritative DNS is operated by two different DNS providers: *AWS Route 53* (DNS server for Amazon Web Services (AWS)) and *Cloudflare*.

Over the course of four months, we sent 109,649 identical emails from the 6 different domains to email accounts hosted by *Microsoft* and *Google*. As we expected that time could play a role concerning the seemingly randomly occurring temp errors in practice (e.g., because the mail volume is higher at certain times of the day, leading to an overload), we sent the test emails every hour. As DMARC reports are received on a daily basis (without detailed times), we chose to use DKIM selectors on a 3-hour basis to be able to investigate the times in more detail. To get straight to the point, time does not matter with regard to the temp errors, as we found.

All emails were sent with proper authentication settings that should be accepted by receiving MTAs:

- **SPF:** The policy "v=spf1 ip4:65.108.248.78 -all" was used for all emails.
- **DKIM:** We used different DKIM selectors to be able to test different RSA key lengths (1024 bit, 2048 bit, and 4096 bit).
- **DMARC:** The policy "v=DMARC1; p=reject; rua=mailto:dmarc@rua.mailtower.app;" was used for all emails.

We placed the DKIM keys directly in the DNS zone of our domain, without a CNAME forwarding as done by Microsoft, for example. This approach has the advantage that the public key does not need to be requested via several DNS servers.

With this setup, we encountered a DKIM temp error rate that matches the situation in practice as seen from the data provided by the DMARC service provider (and being confirmed by the deliverability community as well).

B. The Role of the DNS

During our investigation, we found that it made a difference from which domains emails were sent. Thus, we suspected that the handling of DNS responses by AWS Route 53 and Cloudflare could be different, potentially leading to the seemingly random temp errors during the DKIM check on the receiving mail server side. Fig. 1 shows the results of the investigation.

In the next step, we therefore checked how far AWS Route 53 and the Cloudflare Authoritative DNS Server (ns1.cloudflare.com, ns2.cloudflare.com) behave differently and found that Cloudflare does not include an *Authority-Section* in their DNS responses—in contrast to AWS Route 53. While AWS Route 53 is fully compliant with the "classical" RFCs 1034 [1] and 1035 [12] by including the NS and SOA records in the authority section, Cloudflare foregoes that and stays with the (modern) principle of "minimal responses" from RFC 2308 [13]. This principle is actually thought for negative responses, but large DNS providers like Cloudflare, Google Cloud DNS, etc. use this principle for positive responses as well—mainly for reasons of DNS amplification attack protection. Keeping packet sizes smaller yields better efficiency as fewer fragmentation is needed and stability over UDP is increased. The received bytes shown in Fig. 2[2] indicate that the

[1]https://www.nuget.org/packages/MailKit

[2]Note that the error for DKIM selector 5 is due to the fact that we posed a DNS query without using EDNS. As the response was greater than 512 bytes, we received an error and no response. We did not allow the fallback to TCP in order to test whether the request and response properly work via UDP.

DNS Service Provider	Domain	DKIM Selector	Note	Pass	Temp Error	Fail	Error Rate
Cloudflare	mailtower.app	dkimtestselector4	rsa | 4096	2161	276	0	11,3%
Cloudflare	mailtower.app	dkimtestselector5	rsa | 2048	951	1	0	0,1%
Cloudflare	mailtower.app	dkimtestselector6	rsa | 2048	932	1	0	0,1%
Cloudflare	mailtower.app	dkimtestselector7	rsa | 2048	944	2	0	0,2%
Cloudflare	mailtower.app	dkimtestselector8	rsa | 2048	951	4	0	0,4%
Cloudflare	mailtower.app	dkimtestselector9	rsa | 2048	954	1	0	0,1%
Cloudflare	mailtower.app	dkimtestselector10	rsa | 2048	953	1	0	0,1%
Cloudflare	mailtower.app	dkimtestselector11	rsa | 2048	951	2	0	0,2%
Cloudflare	mailtower.app	dkimtestselector12	rsa | 2048	953	2	0	0,2%
Cloudflare	mailtower.app	dkimtestselector13	rsa | 4096	1098	163	0	12,9%
Cloudflare	mailtower.dev	dkimtestselector1	rsa | 2048	7398	7	0	0,1%
Cloudflare	mailtower.zone	dkimtestselector1	rsa | 2048	3005	4	0	0,1%
AWS Route53	mailtracking.net	dkimtestselector1	rsa | 2048	3087	265	0	7,9%
AWS Route53	mailtracking.net	dkimtestselector2	rsa | 4096	2586	404	0	13,5%
AWS Route53	mailtracking.net	dkimtestselector3	rsa | 2048	2216	217	0	8,9%
AWS Route53	mailtracking.net	dkimtestselector5	rsa | 2048	1246	113	0	8,3%
AWS Route53	mailtracking.net	dkimtestselector6	rsa | 2048	1236	125	0	9,2%
AWS Route53	mailtracking.net	dkimtestselector7	rsa | 4096	1211	148	0	10,9%
AWS Route53	mailtracking.net	dkimtestselector8	rsa | 1024	925	0	0	0,0%
Cloudflare	passworm.com	dkimtestselector1	rsa | 2048	2674	2	0	0,1%
Cloudflare	config.fail	dkimtestselector1	rsa | 1024	2808	2	0	0,1%
Cloudflare	config.fail	dkimtestselector2	rsa | 2048	6761	3	0	0,0%
Cloudflare	config.fail	dkimtestselector3	rsa | 2048	3175	2	0	0,1%
Cloudflare	config.fail	dkimtestselector4	rsa | 4096	2598	123	0	4,5%

Fig. 1. DKIM Temp Errors depending on DNS Service Providers and DKIM Key Lengths (one month observation period).

difference between AWS Route 53 and Cloudflare responses makes up 140 bytes.

It is striking that Cloudflare DNS responses thus do not need EDNS (as the size is $\leq$ 512 bytes) and AWS Route 53 DNS responses do need EDNS—or otherwise, the DNS responses would be truncated or a TCP fallback would be necessary.

We found that if the DNS resolver defines 1232 bytes, both providers respond with a proper response via UDP. Without the EDNS indication, the response is truncated by AWS Route 53. Moreover, all responses via UDP bigger than 2000 bytes are truncated by Cloudflare, while everything properly works via TCP. AWS Route 53, on the other hand, also responds with DNS packets larger than 2000 bytes via UDP.

C. The Role of the Signature Keys

During our investigation, we found that the size of the DKIM keys also plays a role in terms of deliverability. While emails that have been signed with 1024 bit DKIM RSA keys properly work both in the AWS Route 53 and Cloudlflare environment (with zero errors in both cases), longer key sizes yield to problems especially with AWS Route 53.

For example, an RSA DKIM key with a size of 2048 bit yields an error rate of 8-9 % for AWS Route 53 while only 0.1-0.2 % for Cloudflare. This high error rate for 2048 bit RSA keys with AWS Route 53 is especially problematic as 2048 bit RSA keys are state-of-the-art today, with the German Federal Agency for Information Security requiring a key size greater than 3000 bits as of 2025, for example [14].

For a 2048 bit RSA DKIM key, the public key in the DNS needs 410 bytes of payload. This payload fits into 512 byte DNS responses from Cloudflare but not into DNS responses from AWS Route 53 (as the authoritative names servers are included, as discussed in Sect. III-B).

The situation is even worse for DKIM RSA 4096 bit keys. Both AWS Route 53 and Cloudflare yield a 11-13 % error rate for such emails.

The problem is expected to be related to the issue discussed above. In this case, larger key lengths lead to DNS packets exceeding the maximal length of 512 bytes for UDP packets and, thus, running into issues with EDNS.

It is interesting that only emails sent to the Microsoft email account cause problems. Emails sent to the Google account do not yield any problems. Thus, we suspect that Exchange Online (which hosts the receiving mail account) has problems handling EDNS requests.

Domain	Nameserver	EDNS	Received Bytes	Payload	Provider
dkimtestselector9._domainkey.mailtower.app	108.162.195.29	-	482	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	172.64.35.29	-	482	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	162.159.44.29	-	482	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	173.245.58.155	-	482	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	108.162.192.155	-	482	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	172.64.32.155	-	482	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	108.162.195.29	1232	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	172.64.35.29	1232	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	162.159.44.29	1232	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	173.245.58.155	1232	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	108.162.192.155	1232	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	172.64.32.155	1232	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	108.162.195.29	2000	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	172.64.35.29	2000	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	162.159.44.29	2000	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	173.245.58.155	2000	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	108.162.192.155	2000	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	172.64.32.155	2000	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	108.162.195.29	4000	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	172.64.35.29	4000	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	162.159.44.29	4000	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	173.245.58.155	4000	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	108.162.192.155	4000	493	410	Cloudflare
dkimtestselector9._domainkey.mailtower.app	172.64.32.155	4000	493	410	Cloudflare
dkimtestselector5._domainkey.mailtracking.net	205.251.194.233	-	0	0	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.196.85	-	0	0	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.199.228	-	0	0	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.192.211	-	0	0	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.194.233	1232	633	410	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.196.85	1232	633	410	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.199.228	1232	633	410	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.192.211	1232	633	410	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.194.233	2000	633	410	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.196.85	2000	633	410	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.199.228	2000	633	410	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.192.211	2000	633	410	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.194.233	4000	633	410	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.196.85	4000	633	410	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.199.228	4000	633	410	AWS Route53
dkimtestselector5._domainkey.mailtracking.net	205.251.192.211	4000	633	410	AWS Route53

Fig. 2. DNS Response Sizes depending on DNS Provider.

D. The Role of Microsoft

To find a proof for our assumption that Exchange Online has problems handling EDNS requests, we further investigated the Exchange Online infrastructure in more detail.

When an organization sets up Exchange Online, the organization's domain MX record is set to a Microsoft mail server under *mail.protection.outlook.com*. We found that Microsoft establishes a subdomain delegation for mail.protection.outlook.com, which points to other DNS servers. By resolving an organization's DNS record for an email address hosted by Exchange Online, we find that *ns1-proddns.glbdns.protection.outlook.com* is queried. As we found, this DNS server does not support EDNS, though. This missing puzzle piece finally explains why Microsoft customers are faced with the problem of undeliverable emails—due to an inconsistent implementation of EDNS in Microsoft's DNS infrastructure.

As the size of DNS responses turned out to seem to play a role with regard to the DKIM temp errors, we further investigated more aspects of DNS configurations that have an impact on the DNS response sizes—potentially leading to more errors.

E. The Role of too many Verification Entries

Many online services rely on the DNS as a verification mechanism to establish control over a domain name. In particular, domain ownership is commonly proven by publishing service-specific verification tokens as DNS TXT records, which can be queried and validated automatically. As a result, domains often accumulate a large number of TXT entries over time, reflecting interactions with multiple providers for purposes such as certificate issuance, email authentication, and third-party service integration. This widespread practice makes DNS TXT records a rich and densely populated source of ownership and configuration metadata.

During our investigation, we suspected that the large number of such entries might negatively affect SPF verification, as the receiving MTA fetches all TXT records from the DNS of the sending domain to check whether a TXT record for SPF (v=spf1) is available.

A good example of such an overload of the DNS with TXT entries is *adobe.com*. Adobe.com has the following DNS response sizes: A record (70 bytes), AAAA record (94 bytes), MX record (110 bytes), and TXT records (5455 bytes). In total, adobe.com has 69 TXT records; whereby 67 of them are used for domain verification, as can be seen in Fig. 3.

The huge size of the DNS entries for adobe.com (altogether 5729 bytes) leads to the problem that for each received email from adobe.com, the UDP DNS request needs to be switched to a TCP request with a truncate response, which results in a worse performance (as shown in Fig. 4) and a potentially higher error rate. We performed 3 parallel requests, with 10 inquiries each, from 5 different test servers to test the different DNS servers for the domain for our load test. It should be noted that a single received email can trigger around 4-5 DNS requests (FCrDNS, SPF, DKIM, DMARC, BIMI), where alone the SPF validation can trigger several requests in marketing/SaaS setups (with a cap at 10 lookups according to RFC 7208); in the case TCP is needed as fallback, the number of requests doubles.

We can expect an "inexplicable" number of SPF temp errors for emails sent by adobe.com, whose mail provider is Exchange Online as we found, due to the high number of TXT records in the DNS. As Adobe is not a customer of the DMARC service provider that provides the data for this work, we cannot directly verify this.

However, to investigate whether our assumption might hold, we added "artificial" domain verification entries to one of our sending domains (to cross the threshold of 1232 bytes, which is the recommended size agreed upon on the 2019 DNS flag day—as this size shall avoid fragmentation on current networks) in order to replicate the adobe.com case. We found that the *number of SPF temp errors suddenly doubled* from that month on when we added the entries. Thus, we can conclude that the number of TXT entries (above the threshold) indeed increases the risk of undeliverable emails due to SPF temp errors.

F. Interim Conclusion

If an organization has the proper DMARC policy *reject* in place and uses AWS Route 53 as its DNS server, a DKIM key of length 2048 bit or more, and a DNS zone clogged with too

Type	Name	Data	TTL
TXT	@	openai-domain-verification=dv-pN3CfPVdeBAANroxeTFmd7WZ	2744
TXT	@	google-site-verification=MnYkALPA4CNieThUZzzp4Hh88H5szHXokxqirdGFFN0	2744
TXT	@	google-site-verification=Rg8rw6QFcht0wbcstENHTQ3h5l_ujrfBPxhuE7cfBFE	2744
TXT	@	fastly-domain-delegation-rQNX3KD7DKL9hmeR-378696-2021-06-09	2744
TXT	@	google-site-verification=Q86EnJZ_pwCfS8VpA1rlQPS3HXZzrcCWGD68zb74InY	2744
TXT	@	adobe-sign-verification=bb81bc75163acd737f022c8b8ac3958a5f3600ba3daaaa5fad01d44924f21fad	2744
TXT	@	google-site-verification=P_kb2Yyzww7fnZitZ6EbIYipWkjkin9et6IsSwDp7lg	2744
TXT	@	elevenlabs=7WqXlRwQh8-jH2984SP4TQCS0MWL3IoSp8kynyVKVg8	2744
TXT	@	openai-domain-verification=dv-Cf9stelxxYuIRVx3Kd0z28ks	2744

Fig. 3. DNS Snapshot of adobe.com, showing a selection of 10 TXT entries (out of 69) as of Dec. 2025. Cloud services heavily rely on DNS verifications. The number of Google site verification entries in Adobe's DNS zone may indicate an inconsistency of verification across departments using Google services.

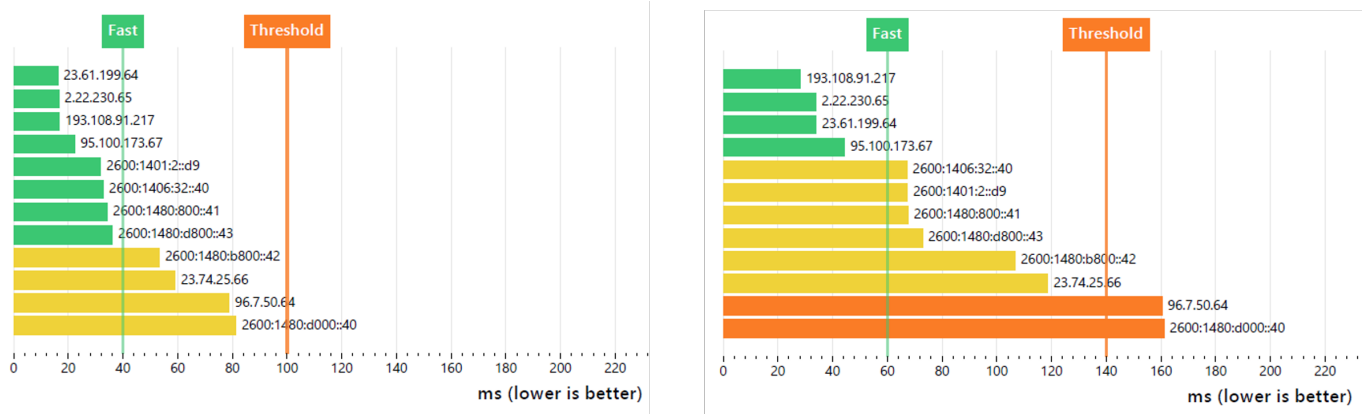

Fig. 4. adobe.com DNS nameserver performance under UDP load 30 (left) and TCP load 30 (right). For emails received from adobe.com, DNS requests over TCP are necessary due to the large size of the DNS response, resulting in a worse performance and potentially a higher error rate.

many TXT entries, the chances are high that a number of their e-mails sent to Exchange Online accounts are undeliverable as DKIM and SPF temp errors are expected to occur—and the reject policy requires at least the SPF or the DKIM validation to be successful (in addition to proper alignment).

G. The Role of Ed25519-SHA256

The *German Federal Agency for Information Security* (BSI) states in its technical guideline "BSI TR-03182 Email Authentication" from 2024 [15] that Ed25519-SHA256, as specified in RFC 8463 [16], shall be used for signing outgoing emails with DKIM and receiving MTAs *must* support its verification. The BSI also states in the guideline: "Nevertheless, ED25519-SHA256 is not gaining the widespread adoption it should, even though there is much to be said for its use: the length of the public key is limited to 256 characters, it fits into a single TXT record, leads to fewer implementation issues, allows querying over the faster UDP rather than the slower TCP protocol, and to top it off the algorithm provides more cryptographic security." [15]

As part of our investigation, we checked whether email providers use Ed25519-SHA256 for DKIM signing, but we could not find a single email provider that does that. Furthermore, we checked whether Exchange Online and Google Mail support DKIM verifications for incoming emails (which is a must requirement according to the BSI). However, we found that both email providers do not support the verification of Ed25519-SHA256 signatures and reject such emails. If both, an Ed25519-SHA256 signature and an RSA signature as fallback are used, Microsoft sends a DMARC report without any information about DKIM—leaving the sender "blind" with regards to deliverability information.

IV. Possible Solutions

In this section, we present possible solutions to the found problems.

A. Clearing out the DNS Zone

As we have seen in Sect. III-E, the exuberant use of the DNS for domain verification purposes affects the verifiability of SPF. Even though this is primarily a problem for recipients with Exchange Online accounts, the whole practice should be reviewed. From what we have found in the wild, especially big companies (like Adobe), with a huge number of such verification TXT records, are affected. If these companies face the actual problem of certain authenticated emails being rejected for no obvious reason, they should take steps to mitigate this DNS zone overload. A possible solution could be to put the verification records into subdomains, as they would not affect SPF checks in this case, or to use a CNAME entry.

Another possible solution could be that AWS Route 53 does not include the authoritative name servers in their DNS responses, which also contributes to smaller responses (and fewer problems).

B. DKIM signing algorithms and key sizes

As we have found, emails signed with "too long" DKIM signature keys lead to deliverability problems. In practice, this problem might not have been dramatic in the past, as MTAs signed outgoing emails with 1024 bit RSA keys, from what we have seen in the data. Today, 2048 bit RSA keys constitute the de-facto standard in practice, leading to problems in some cases already, as we have seen. If key lengths are further increased, the error rate increases as well. However, relying on short keys to prevent the problem of rejected emails due to failures is not the best solution from a security point of view.

Moreover, MTAs shall also start supporting Ed25519-SHA256. One explanation for the reluctance to support Ed25519-SHA256 from our point of view is that the computational overhead for the signer tremendously decreases with Ed25519-SHA256. Compared to RSA with a key size of 4096 bit, which takes 22 ms for signing a 7 KB email, signing the same email only takes 0.2 ms with Ed25519-SHA256—while the needed computational effort for verification approximately stays the same. Thus, the fear might be that a decrease of computational overhead for senders might increase the potential for misuse, as spammers could sign emails with fewer overhead.

It should be noted that, as stated in the technical guideline of the BSI [15], Ed25519-SHA256 avoids the need to split the TXT records (as is the case for RSA 2048 and RSA 4096 bit keys) due to the lower key size.

C. Exchange Online

As shown in Sect. III-B, Exchange Online seems to have a problem in posing DNS requests if the responses are too large and EDNS would be necessary. This especially concerns email senders whose DNS zone is managed by AWS Route 53, whose DNS zone contains too many entries, or whose DKIM key size is "too long" (even if a state-of-the-art key size is used). It should be straight-forward for Microsoft to solve these issues by properly setting up EDNS, which is supported since Microsoft Windows Server 2008.

Microsoft does not support DANE for *mail.protection.outlook.com*. If a customer wants to set up DANE for an Exchange Online-hosted email account, he needs to set up DANE with laborious effort via Powershell (i.e., it is not possible to set up DANE via the web interface). We found that when a customer sets up DANE, the domain *mx.microsoft*, which is the *new* delivery domain (introduced in March 2024) for Exchange Online, is used instead of mail.protection.outlook.com. And the new delivery domain in fact supports EDNS. Thus, Microsoft should use the new delivery domain for all customers in the future, to avoid the problems as uncovered in this paper and to further increase DANE usage (which also lacks behind [17]) in practice as well.

D. Support for Ed25519-SHA256:

If a sending MTA uses the Ed25519-SHA256 signature scheme to sign outgoing emails with DKIM (as recommended by the BSI [15]), both Exchange Online and Google Mail as the receiving MTAs will reject the emails due to a "DKIM fail: syntax error" in all cases. Again, the solution would be for email providers to introduce support for Ed25519-SHA256 in the near future.

V. Related Work

Early empirical studies on email authentication—particularly SPF, DKIM, and DMARC—revealed substantial gaps in deployment and enforcement across the email ecosystem. Durumeric et al. [18] and Foster et al. [19] conducted some of the first large-scale measurements, showing that while SPF and DKIM were already standardized, adoption remained low and enforcement was inconsistent across providers. These works also demonstrated that many providers either ignored authentication failures or treated them as soft signals, resulting in limited protection against spoofing.

Subsequent studies refined this picture. Research such as Maroofi et al. [20] used DNS-based evaluations to show widespread SPF misconfigurations, often due to overly permissive or syntactically invalid records. Deccio et al. [21] examined sender validation behavior via triggered DNS lookups and found that although adoption had improved, a large portion of mail servers still failed to check all three mechanisms consistently. Wang et al. [22] conducted the first longitudinal analysis of DKIM, identifying issues such as long-lived keys and misconfigurations in selectors, and confirming that many domains continue to rely on outdated or weak cryptographic parameters.

The Extended Hell(o) study by Blechschmidt et al. [23] from 2023 provides the most holistic and provider-focused analysis to date. By performing controlled outbound tests against 47 major mail providers and DNS-based scans over the top 10 million domains, the authors replicate and significantly extend earlier work. Their results show that:

- SPF deployment has stagnated: Only 40% of domains publish SPF records, essentially unchanged from 2015, and 4,183 domains even explicitly authorize arbitrary senders via misconfigured "+all" or "all" mechanisms.
- DKIM adoption among providers has improved, with all 47 tested providers querying DKIM selectors—compared to only half in 2015—but enforcement remains inconsistent, and 14 providers still accept messages with invalid DKIM signatures.
- DMARC adoption has increased, but enforcement lags: 37 providers publish DMARC records, but only 25 act on failures. The study also uncovers RFC-noncompliant handling of subdomain policies by two providers.

The large-scale DNS measurements further reveal widespread real-world misconfigurations: 371,968 SPF records produce permanent errors, and 9,873 DMARC policies contain syntax errors that invalidate their intended protection. These findings highlight a persistent gap between specification and operational practice, underscoring that despite years of standardization efforts, SPF, DKIM, and DMARC remain only

partially—and often incorrectly—deployed. In contrast to prior work, this study provides the first provider-behavior–oriented comparison across SPF, DKIM, and DMARC, evaluating how major MTAs react to intentionally malformed or conflicting records. Its results show that even well-known "secure" providers do not reliably enforce authentication failures, enabling practical sender-spoofing attacks in 28 of 47 tested services. This expands earlier insights into authentication weaknesses by showing that the problem is no longer just one of adoption, but of operational enforcement quality across the email ecosystem.

VI. Conclusion and Outlook

By working together with a DMARC service provider who has insights into a huge number of clients' reports on email deliverability issues, we could set up a test environment in order to investigate the question why more and more properly *authenticated* emails are rejected in practice due to so far seemingly random temp errors. We are the first to be able to pinpoint the problems that have been actively discussed in the deliverability community in the past months—since enforcement of email authentication with SPF, DKIM, and DMARC by the big email providers has been increased. Our results thus present solutions to a timely problem in practice.

We show that the problems are manifold but can be mainly nailed down to the underlying infrastructure supporting the email authentication: the DNS. Not only does it matter how DNS responses are constructed by DNS providers (e.g., if they include an authoritative section), but also the fact that DNS zones are crowded with TXT entries serving for domain verification purposes has an impact on SPF validation and, thus, email deliverability. Regrettably, "higher security" (with state-of-the-art DKIM key lengths and modern signing algorithms) also leads to a higher email rejection rate.

Finally, we found that Microsoft seems to have a problem with the support for EDNS in Exchange Online, its popular cloud-based mail service used by many organizations to handle their emails. EDNS has a 25 year history and was introduced in Windows Server 2008. While EDNS is not mandatory, it is "quasi-duty" since around 2010 when DNSSEC got productive and DNS zones got bigger. This lack of support further exacerbates the problem of increasing mail rejections.

References

[1] P. Mockapetris, "Domain Names - Concepts and Facilities," RFC 1034, 1987. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1034

[2] R. Barnes, J. Hoffman-Andrews, J. McCarney, and D. Kasten, "Automatic Certificate Management Environment (ACME)," RFC 8555, 2019. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8555

[3] M. Wong and W. Schlitt, "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1," RFC 7208, 2014. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7208

[4] M. Kucherawy, "DomainKeys Identified Mail (DKIM) Signatures," RFC 6376, 2011. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6376

[5] M. Kucherawy and E. Zwicky, "Domain-based Message Authentication, Reporting, and Conformance (DMARC)," RFC 7489, 2015. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7489

[6] S. Dickinson, V. Rijswijk-Deij, M. Kosek, and A. Mankin, "DNS Query Name Minimisation to Improve Privacy," RFC 9156, 2021, appendix A discusses trends in DNS TXT record usage. [Online]. Available: https://www.rfc-editor.org/rfc/rfc9156

[7] J. Damas, M. Graff, and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))," RFC 6891, 2013. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6891

[8] DNS-OARC, "DNS Flag Day," https://dnsflagday.net, 2025, accessed 2025-12-01.

[9] Google, "Email Sender Guidelines Update 2024," https://workspace.google.com/blog/security/email-sender-guidelines, 2024, accessed 2025-12-01.

[10] Yahoo, "Email Sending Requirements 2024," https://senders.yahooinc.com/postmaster/requirements, 2024, accessed 2025-12-01.

[11] Microsoft, "Microsoft Email Authentication Roadmap," https://learn.microsoft.com/en-us/microsoft-365/security/office-365-security/email-authentication, 2024, accessed 2025-12-01.

[12] "Domain names - implementation and specification," RFC 1035, Nov. 1987. [Online]. Available: https://www.rfc-editor.org/info/rfc1035

[13] M. P. Andrews, "Negative Caching of DNS Queries (DNS NCACHE)," RFC 2308, Mar. 1998. [Online]. Available: https://www.rfc-editor.org/info/rfc2308

[14] Bundesamt für Sicherheit in der Informationstechnik (BSI), "Technische Richtlinie 02102-1: Kryptographie Verfahren: Empfehlungen und Schlüssellängen," 2025.

[15] ——, "Technische Richtlinie 03182: E-Mail Authentication," 2024.

[16] J. Levine, "A New Cryptographic Signature Method for DomainKeys Identified Mail (DKIM)," RFC 8463, 2018. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc8463

[17] C. Lange, T. Chang, M. Fiedler, and R. Petrlic, "An email a day could give your health data away," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, J. Garcia-Alfaro, G. Navarro-Arribas, and N. Dragoni, Eds. Cham: Springer International Publishing, 2023, pp. 53–68.

[18] 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 *Proceedings of the 2015 ACM Internet Measurement Conference (IMC)*, 2015.

[19] 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 *Proceedings of the 2015 ACM Conference on Computer and Communications Security (CCS)*, 2015.

[20] S. Maroofi, M. Korczynski, A. Hölzel, and A. Duda, "Adoption of email anti-spoofing schemes: A large scale analysis," *IEEE Transactions on Network and Service Management*, 2021.

[21] C. Deccio, T. Yadav, N. Bennett, A. Hilton, M. Howe, T. Norton, J. Rohde, E. Tan, and B. Taylor, "Measuring email sender validation in the wild," in *Proceedings of ACM CoNEXT*, 2021.

[22] C. Wang, K. Shen, M. Guo, Y. Zhao, M. Zhang, J. Chen, B. Liu, X. Zheng, H. Duan, Y. Lin, and Q. Pan, "A large-scale and longitudinal measurement study of dkim deployment," in *USENIX Security Symposium*, 2022.

[23] B. Blechschmidt and B. Stock, "Extended hell(o): A comprehensive large-scale study on email confidentiality and integrity mechanisms in the wild," in *Proceedings of the USENIX Security Symposium*, 2023.