IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 25, 2020, accepted July 8, 2020, date of publication July 14, 2020, date of current version July 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009122

What Email Servers Can Tell to Johnny: An Empirical Study of Provider-to-Provider Email Security

GEORGIOS KAMBOURAKIS, GERARD DRAPER GIL, (Member, IEEE), AND IGNACIO SANCHEZ, (Member, IEEE)
European Commission, Joint Research Centre (JRC), 21027 Ispra, Italy

Corresponding author: Georgios Kambourakis (georgios.kampourakis@ec.europa.eu; gkamb@aegean.gr)

ABSTRACT With hundred billions of emails sent daily, the adoption of contemporary email security standards and best practices by the respective providers are of utmost importance to everyone of us. Leaving out the user-dependent measures, say, S/MIME and PGP, this work concentrates on the current security standards adopted in practice by providers to safeguard the communications among their SMTP servers. To this end, we developed a non-intrusive tool coined MECSA, which is publicly available as a web application service to anyone who wishes to instantly assess the security status of their email provider regarding both the inbound and outbound communication channels. By capitalising on the data collected by MECSA over a period of 15 months, that is, ≈7,650 assessments, analysing a total of 3,236 unique email providers, we detail on the adoption rate of state-of-the-art email security extensions, including STARTTLS, SPF, DKIM, DMARC, and MTA-STS. Our results indicate a clear increase in encrypted connections and in the use of SPF, but also considerable retardation in the penetration rate of the rest of the standards. This tardiness is further aggravated by the still low prevalence of DNSSEC, which is also appraised for the email security space in the context of this work.

INDEX TERMS Email security, Internet measurement, network security, SMTP.

I. INTRODUCTION

Email has its technological roots in the pre-Internet era with the ratification of RFC 821, namely, the Simple Mail Transfer Protocol (SMTP) in the early 80's. The later on standardization of HTTP and the advent of the Internet in the 90's, led SMTP to become one of the cornerstone protocols supporting the modern worldwide Internet service infrastructure. In the meantime, the Internet has proved to be a fertile ground for the proliferation of an endless number of digital services that have revolutionised virtually every aspect of public and private life. In line with this evolution, today, email is still massively used [1] as a traditional communication channel to complement the current ecosystem of modern similar services, including the plethora of mobile messengers and chat apps.

On the other hand, the increased dependency on online services and the augmenting number of cyber threats constantly raise a number of security and privacy concerns. The same concern applies to email communications given that email messages either personal, business, or e-government oriented often contain personal data, and email addresses are extensively used for conducting digital identity management, say, for fallback authentication. Recently, the EU's General Data Protection Regulation (GDPR) [2], highlights this need stating that when personal data is exchanged over email, appropriate technical and organisational measures must be put in place to ensure the confidentiality and integrity of the relevant processing systems and services. Moreover, email-borne attacks are prevalent and constantly evolving over time; email is an ordinary social engineering channel, and it is widely exploited by scammers, hackers, and identity thieves, especially in times of crisis, such as in the current COVID-19 pandemic. On top of that, email is a common method for spreading malware using deceptive messages to lure recipients to click on seemingly innocuous hyperlinks or attachments. It is therefore straightforward that effective mitigation of security, privacy and data

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

protection risks in email communications is of paramount importance.

In fact, whereas other key Internet protocols, such as HTTP and its evolution into HTTPS, have received a considerable amount of attention by both the industry and the academia, SMTP [3] has not been subjected to the same degree of scrutiny. This is because security is bolted on top of email entirely as an afterthought. By all odds, secure end-to-end email communication still remains an issue more than 19 years after Whitten and Tygar's seminal paper, ''Why Johnny Can't Encrypt'' [4].

Precisely, for the sake of gradual deployment of modern SMTP security extensions, the email service follows a ''fail-open'' model, and thus offers no assurances regarding confidentiality, integrity, and authenticity. This situation directly affects the communication link between an end-user and the corresponding SMTP server as well as that among SMTP servers, which is the focus of this work. For end-user to email server communications, RFC 8314 [5] deprecates the use of cleartext. In addition, end-users can rely on the OpenPGP [6] and S/MIME [7] standards to achieve the aforementioned security goals in an end-to-end fashion. Even so, the reality shows that the adoption rate of these two competing standards remains low mainly due to usability issues [8]. In short, especially for communications happening between SMTP servers, the ordinary end-user needs to blindly trust their email provider, that is, without having a simple way of checking whether the provider correctly implements and imposes the latest email security standards at its end.

Our contribution: The work at hand focuses on the communication between SMTP servers, and addresses the email security malady from both an end-user's and internet measurement viewpoint. First, we develop a simple-to-use public service coined ''My Email Communications Security Assessment'' (MECSA) [9] for enabling a user to check instantly and at any time the security level of their email provider in a voluntary, privacy-preserving, non-intrusive manner. Second, based on the wealth of data collected by MECSA over a time period spanning from Jan. 2019 to Mar. 2020, we portray the global penetration rates of the current SMTP security extensions to the respective providers. The derived results vis-à-vis those reported by the relevant but scarce literature offer a holistic view of the adoption level of these technologies over time. As a side contribution, and for the sake of spurring further research on this topic, we release as open-source a command line version of the MECSA service engine called ''MECSA-ST'' [10].

The remainder of the paper is organized as follows. The next section succinctly presents the architecture and protocols of the email ecosystem. Section III details on our methodology, while section IV presents and elaborates on the results of the security assessments conducted by MECSA. The related work is discussed in Section V. Finally, Section VI concludes and provides pointers to future work.

II. BACKGROUND

E-mail systems are based on a client-server architecture. An email is sent from a client to its local SMTP outgoing server, which relays the email to its intended destination, that is, the local incoming SMTP server of the receiver's domain. In practice, due to mail forwarding, email lists, and internal mail processing within an organisation, an email may be relayed through several SMTP servers before reaching its final destination.

The email infrastructure comprises two basic software agents, namely *Mail User Agent (MUA)* and the *Mail Transfer Agent (MTA)*. The MUA is used to send (push) and receive (pull) emails at the user side, i.e., a real person or another application. The email message is received by a server program called the *Mail Submission Agent (MSA)*, it is checked, say, for errors and transferred typically via SMTP to the MTA. The MTA is the process within an SMTP server that takes care of receiving emails, either from the MSA, or another MTA, and delivering them, either to another MTA or the *Mail Delivery Agent (MDA)*. The latter entity, which is also known as the *Local Delivery Agent*, filters and possibly stores the email message into the mailbox. For relaying the email to its intended destination, the outbound MTA is also required to locate the Mail Exchanger (MX) Resource Record (RR) corresponding to the recipient's network domain DNS zone. Given that a MX RR points to a server, say, *smtp.destination.eu*, the outbound MTA needs to also trigger another DNS query for finding the destination server's IP address. Communications within the same provider, e.g., between the MSA and MTA can be deemed trusted because these agents are usually co-located, e.g., in the same host and protected by physical means. Nevertheless, for large providers this may not be the case, and measures to protect intercommunication are required. Given that, in the following sections, the term ''MTA'' is used to generally refer to all server-side operations.

Email is built over a set of three core protocols. SMTP [3] is used in the communications between MTAs, and between MUAs and MTAs when sending emails. It runs over TCP on port 25 for MTA-to-MTA communications and on port 587 for MUA-to-MTA relaying. SMTP was introduced as an ASCII only based protocol and later updated with the Multipurpose Internet Mail Extensions (MIME) [11]–[15], and the definition of a flexible service extension model, the Extended SMTP (ESMTP), which introduced advanced features, including STARTTLS [16]. Post Office Protocol v3 (POP3) [17] and Internet Message Access Protocol (IMAP) [18] are used in communications between MUAs and MTAs when accessing emails.

A. SECURITY ASPECTS

The scope of the present work is limited to the security provisions regarding MTA-to-MTA communications. We are particularly interested in whether, how, and up to what degree the email providers safeguard specific security properties,

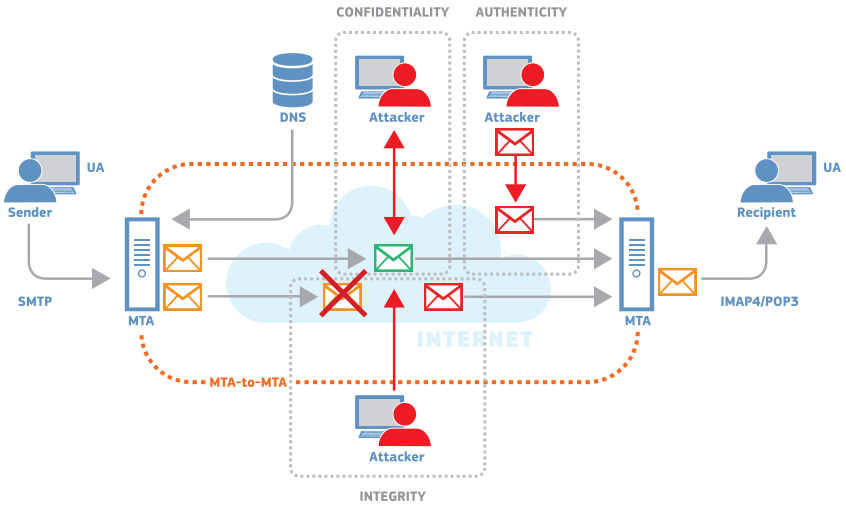

FIGURE 1. High-level depiction of attacks against email confidentiality, authenticity, and integrity.

namely message confidentiality, authenticity, and integrity when interacting with other providers. Naturally, this necessity is tightly connected with the level of trust an end-user can place in their provider. This is simply because the ordinary end-user cannot straightforwardly be informed about the security practises followed by their provider, and indeed the current work provides a solution to this need. In reality, as it is elaborated in the subsequent sections, the three aforementioned security properties are hardly satisfied in their entirety by a large mass of providers. For instance, the SMTP protocol does not require MTAs to authenticate, assuming that every email received is legitimate. Therefore, the repertoire of network attacks available to the passive or active opponent, range from eavesdropping on the communication channel or man-in-the-middle (MITM), to setting up a rogue MTA to launch spam or phishing campaigns. The next paragraphs along with figure 1 briefly analyse and give a high-level representation of attacks against the security properties of interest.

1) *Confidentiality:* Differently from HTTPS, email offers opportunistic encryption over TLS on a hop-by-hop fashion. This fail-open model potentially enables passive eavesdroppers to snoop on the communications between MTAs. An active adversary is also able to strip or distort the announcement of TLS to force the receiving end to fall back to cleartext. Even if a TLS tunnel is created, but the outbound MTA does not authenticate the inbound by means of server certificate validation, then an active attacker can act as a MITM impersonating each end of the connection to the other. This would allow the aggressor to view and tamper with any message at will.
2) *Authenticity:* The inbound MTA should verify the outbound MTA appearing in the "Received" header of the message vis-à-vis the sender's email domain appearing in the "From" header. Put simply, is the provider of the inbound message authorised to send emails on behalf of the original sender? Otherwise, an evildoer is able to successfully inject a forged message anywhere in the MTA-to-MTA email path.
3) *Integrity:* The preservation of the integrity of email messages, including the sender's address, in transit between MTAs cannot be guaranteed if appropriate measures like digital signatures are not enforced. Plainly, it is infeasible to assure that the message received is identical to the original; an active attacker is able to manipulate both the content of the message and the associated metadata, such as the sender and recipient along any hop between the involved MTAs.

Given the above observations, we define the following *adversary model*: Adversaries are individuals, groups, or organizations who attempt to compromise the security of the email service by specifically aiming at MTA-to-MTA network hops. We consider passive or active adversaries with the following capabilities: (a) they can intercept, block, modify, or inject any message in the public communication channel; (b) they adhere to all cryptographic assumptions, e.g., an adversary is unable to decrypt an encrypted message without knowing the decryption key; (c) they are neither in position to compromise an existing legitimate mail server, nor interfere with the communication links between email entities within the same email provider, but they are able to setup and operate their own server; (d) they are able to attack the DNS protocol, but not, say, tamper with DNSSEC-protected responses due to the previous point (b).

B. EMAIL SECURITY STANDARDS

The present work concentrates on the following security standards pertaining to the email ecosystem.

- The *STARTTLS* extension [16], [19] to SMTP allows the use of TLS in communications between SMTP clients and servers. Specifically for MTA-to-MTA

communication, every inbound MTA that supports this extension announces it by including the keyword "STARTTLS" in the response to the SMTP Extended Hello ("EHLO") command issued by the outbound MTA. To proceed with the negotiation of a TLS tunnel, the outbound MTA will respond by issuing a "STARTTLS" command toward the receiving end. Assuming the use of a strong ciphersuite, STARTTLS mitigates attacks on confidentiality. In an effort to prevent MITM attacks, X.509 certificates are traditionally used to validate the identity of SMTP servers during the TLS handshake. The rules applicable to email server certificate verification are specified in [19]. The willingness of the parties to employ TLS must not be considered a panacea because as indicated in subsection II-A, point (a), downgrade attacks are feasible by tampering with the establishment of the TLS session. It is emphasised that RFC 7817 [19] does not apply to MTA-to-MTA communications. That is, as per RFC 8314 [5] recommendation, TLS with SMTP for message relay should involve either DANE or MTA-STS, as they are discussed in the following.

- *Sender Policy Framework (SPF)* [20] is a protocol that allows email providers to announce a list of hosts authorized to deliver emails on its behalf. SPF records are published as TXT type RR in the DNS zone of the email provider, and thus directly depend on the integrity of DNS. Only one SPF TXT RR per domain is allowed, but this record can list multiple authorised servers. As an email authentication technique, SPF is an effective measure to block the delivery of messages from unauthorised or dubious sources, and therefore reduce unsolicited bulk email.
- The *DomainKeys Identified Mail (DKIM)* [21] standard allows the receiving MTA to validate the origin and contents of an email. DKIM uses digital signatures to bind the email message with its origin, i.e., the holder of the corresponding private key. An email provider supporting DKIM holds one or more private keys, and publishes their associated public keys as DNS TXT RR with the URL < *selector*>._*domainkey.*< *domain*>, where the *selector* parameter is an arbitrary string chosen by the sender. The "selector" and the "domain" parts are included in the signature, respectively "s =" and "d =", to guide the recipient to locate the corresponding public key in the DNS. This also means that a domain can have as many DKIM public keys - along with the corresponding DNS TXT RR stored in the subdomain "_domainkey" - as MTAs that send and sign email; however, each key must use a different selector value. Given that, as with SPF, DKIM also relies on the integrity of DNS. The application of DKIM is an effective measure to mitigate attacks against authenticity and integrity.
- *Domain-based Message Authentication, Reporting and Conformance* (DMARC) [22] is a scalable mechanism that allows the outbound email provider to indicate that its messages are protected by SPF and/or DKIM, and publish policies that inform the receiving end about what to do (none, quarantine, reject) if these authentication methods fail, and how to report related incidents. DMARC policies are published in the DNS as TXT RR, and thus should be secured by DNSSEC as well. Only one RR of this kind is allowed per provider. For instance, the RR "v = DMARC1;p = quarantine;pct = 100;rua = mailto:postmaster@test.org" asks the inbound MTA to quarantine ("p =") all invalid messages and send an aggregated report to the specified address ("rua ="). The optional parameter "pct =" indicates the percentage of messages from the specific mail stream subjected to filtering, hence allowing a gradual deployment of SPF and DKIM. All in all, DMARC, along with SPF and/or DKIM, is an effective weapon in the fight against fraudulent email.
- *Domain Name System Security Extensions (DNSSEC)* [23]–[25] comprises a suite of specifications that basically provide the means for authenticating DNS records. The use of DNSSEC assures that DNS RRs are valid and have not been modified or tampered with, increasing the level of trust. Bearing in mind that the integrity of SPF, DKIM, and DMARC DNS TXT RRs depend on the security of the underlying DNS infrastructure, DNSSEC is the basis for secure email transmission.
- *DNS-based Authentication of Named Entities (DANE)* [26]–[29] is a mechanism that binds certificates to domain names. That is, rather than depending on Certification Authorities (CA), DANE relies on DNSSEC for publishing public keys and certificates for use during TLS handshake. This is done via a specific type of DNSSEC-validated RR, coined TLSA RRs, say, *_25._tcp.mail.test.org. IN TLSA 3 0 1 < digest>*, where the TLSA certificate usage, i.e., how to verify the certificate, is DANE-EE(3), the selector is Cert(0), and the matching type is SHA2-256(1). DANE alleviates two basic problems; (a) the often unclear relationship between an incoming email server domain and the authoritative SMTP server for that domain, and (b) the fact that for trusting the certificate presented by the incoming server, the sender must trust a large number of CAs, but thus far, no universally agreed list of trusted CAs exists. Altogether, DANE makes a TLS connection less vulnerable to downgrade and MITM attacks. It is to be noted that as per RFC 7672 [29], MTA-to-MTA communication should not use TLSA RRs with PKIX validation, namely types PKIX-TA(0) or PKIX-EE(1) [27], and servers that do not support TLS must not publish TLSA records. Lastly, according to RFC 6698, multiple TLSA RRs can be published per host name, e.g., in the case of certificate rollover.
- *SMTP Mail Transfer Agent Strict Transport Security (MTA-STS)* [30] is a mechanism to publish policy directives regarding the use of TLS connections and X.509

certificate validation. MTA-STS is developed as an alternative to the use of DANE, considering that the deployment of DNSSEC is not straightforward and it can be error-prone [31]–[35] and sometimes impractical. That is, differently to DANE, MTA-STS relies on CAs (PKIX) and does not mandate DNSSEC. A policy must be uploaded to the ''.well-known'' location on a web server called the ''Policy host'', e.g., for the domain test.org, in https://mta-sts.test.org/.well-known/mta-sts.txt. The policy *mode* can be ''enforce'', ''testing'', or ''none'', where enforce means that a ''sending MTAs must not deliver the message to hosts that fail MX matching or certificate validation or that do not support STARTTLS'' [30]. The policy of the recipient server is fetched by a sending MTA via HTTPS, thus the policy host certificate must be valid and publicly-trusted. Also, to notify support of MTA-STS, the email provider has to add two DNS RRs; a TXT record, using the *_mta-sts* hostname prefix, and an A or CNAME pointing to the policy host. As per RFC 8461, if multiple valid TXT records for ''_mta-sts'' are returned by the resolver, the sending end must assume that the inbound domain does not support MTA-STS.

III. METHODOLOGY

The current section details on our methodology to assess the level of protection offered by email providers regarding MTA-to-MTA communications. That is, with reference to the standards of the previous section, the objective here is to identify which of them are supported by the evaluated email provider. Our goal is not only to acquire a snapshot of the current penetration rate of these standards to the email providers worldwide, but also to make available the underlying security assessment process to the interested end-user, even the non-tech-savvy one.

To this end, we designed a set of non-intrusive and automated security evaluation tests and combined them into the MECSA web service [9]. By ''non-intrusive'' we mean that these tests do not actively probe the target domains, say, when testing STARTTLS we avoid negotiating all possible ciphersuites. Specifically, we do not overburden the examined email domain by either deliberately sending it a surge of either well-formed, malformed, or some policy-violating emails to exhaustively test all of its receiving MTAs (inbound channel) or forcing the domain to send us numerous emails - at least one per its available MTA - for testing the outbound channel. We do however gather information about the different email servers a given domain may operate because: (i) for the inbound channel, each domain evaluation request submitted to MECSA by a user sparks off a series of STARTTLS-oriented tests against all the available MTAs of that domain plus a small number of inoffensive queries to the DNS as explained in subsection III-A, and (ii) for the outbound stream, the sending MTA of that same domain across different user requests might be different. This means that MECSA manages to evaluate all the available MTAs of a certain domain in a non-intrusive, mostly passive manner, although this typically requires more time to be fulfilled for the outbound channel; it actually depends on how many users of the same email domain will submit evaluation requests to MECSA and reply to the received email.

The implemented tests consider both the communication channels from an MTA viewpoint, namely inbound and outbound. The first one evaluates the reception of emails, whereas the latter assesses their delivery. StartTLS, apply to both directions. SPF, DKIM, DMARC also apply to both directions, but it will be only fully considered in the outbound channel. This is because testing the inbound channel would require an intrusive approach, e.g., sending a set of emails, some complying with the SPF or DKIM policies and some deliberately malformed, and waiting for the server's answer. So, in these cases, we will assume that if an email service applies DMARC, SPF, and DKIM in the outbound channel, it also will impose them in the inbound channel.

A. TESTS

1) INBOUND CHANNEL

Checks performed against the receiving MTA.

First, the prioritised list of DNS MX RRs of the receiving domain is obtained. If the domain does not have at least one MX RR, we use the same domain as MX, i.e., the A type DNS RR is used. Then, exhaustively for each MX host in the RR, we test if it offers the STARTTLS extension by initiating an SMTP connection and sending the ''*EHLO*'' command. If positive, we trigger a TLS handshake to download the certificate along with the intermediate certificate chain, if any. Last, the following checks per certificate are executed [36], [37]:

1) Full Qualified Domain Name (FQDN): Is the MX hostname present in the certificate's Subject Alternative Names (subjectAltName) extension? If false, an additional check is done against the relative distinguished name (CN-ID) contained in the certificate's subject field [36].
2) Expiration Date: The current date is compared against the expiration date shown in the certificate. This check also applies to DANE-TA(2) type of certificates.
3) Revocation List: The Certificate Revocation List (CRL) indicated by the certificate is fetched and checked against the certificate's serial number. OCSP-based revocation checking is to be added in a subsequent version of the platform. Note that as per RFCs 6698 [26] and 7671 [28], revocation for DANE type of certificates relies on DNSSEC.
4) Certificate Authority (CA): The signatures of all the certificates in the chain up to the CA authority are validated. This is done using the list of trusted CAs from an Ubuntu 16.04 distribution. Especially for DANE-TA(2) type of certificates, this requires that the issuing CA certificate (trust-anchor) must be configured in the MTA's certificate chain file.

To look over MTA-STS support, we first query the DNS for the existence of a TXT RR with the *_mta-sts* hostname prefix, and an A or CNAME RR pointing to the policy host. In case of a positive result, the corresponding policy is fetched and examined for possible errors. According to RFC 8461, ''MTA-STS is designed not to interfere with DANE deployments when the two overlap; in particular, senders who implement MTA-STS validation must not allow MTA-STS policy validation to override a failing DANE validation''. That is, due to its reliance on the ''trust on first use'' model and its susceptibility to attacks targeting policy discovery [30], MTA-STS is generally considered less secure than DANE. Given that, if an email sender realises that the recipient supports both DANE and MTA-STS, it should ignore the MTA-STS and process the delivery according to the DANE protocol.

DANE is examined for each MX host advertised by the receiving domain. First, we check the existence of a TLSA RR by issuing a query to *_25._tcp.somedomain*, and if it exists, we try to match the certificate obtained from the MX host with the information downloaded from the TLSA RR.

Regarding DNSSEC, our objective is to examine if the email service offered by the receiving provider is protected by DNSSEC, i.e., the MX RRs, the TXT and TLSA RRs (if applicable), and the A RR of each MX host afford DNSSEC protection. The following inspections are executed:

1) Test if the receiving domain is DNSSEC-enabled. We check if there is a Delegation of Signing (DS) RR in the parent domain, and if so, we validate the whole keychain of pairs DNSKEY/KSK.
2) Test if the MX RRs are protected with DNSSEC; this requires the validation of the signature of the domain's MX RRs.
3) Test if the MX hosts are DNSSEC-enabled. In case the hostname of the MX RRs belongs to a different domain, step (1) is repeated for each host in the MX records.
4) Test if the A RR of the MX hosts is protected with DNSSEC; this requires the validation of the signature of the domain's A RRs.
5) Test if the TXT RRs of the domain are protected with DNSSEC. This step is only required if either SPF, DKIM or DMARC are supported.
6) Test if the TLSA RR of the MX hosts are protected with DNSSEC. This step is only done if DANE is supported.

In sum, we consider that an email provider does support DNSSEC if DNSSEC is enabled both in the domain and the MX hosts, and the MX records are correctly signed, and the A RRs per MX are correctly signed, and the TXT and TLSA RRs are signed.

The support of SPF, DKIM, and DMARC is not examined in the inbound channel. As already pointed out, this would necessitate an intrusive approach in which, for each test, we should send several emails to the tested domain including some purposefully forged to fail the tests. Moreover, to be thorough, it would require sending the same set of emails to each MX host. Therefore, for these three standards, it is assumed that a domain supports them if it passes the outbound test.

2) OUTBOUND CHANNEL

Checks against the sending MTA.

The STARTTLS test in the outbound direction examines if the sending MTA has negotiated the establishment of a TLS encrypted tunnel. To test the support of SPF, we first check the corresponding TXT DNS RR for the domain, i.e., *_spf.<test.org>*, where test.org is the domain tested. If such an RR exists and it is syntactically correct, we use the Python library *pyspf* [38] to validate the email received against the obtained SPF RR. To test for DKIM, we feed the entire message to the *dkimpy* Python library [39] To test DMARC, we look for the existence of a matching TXT RR, i.e., *_dmarc.test.org*, and examine if the latter is syntactically correct.

X.509, DANE, and MTA-STS are not verified on the outbound channel, as this would require an intrusive tactic. That is, since we need to receive an email to inspect the outbound channel, testing these standards would impose an interactive approach, necessitating the reception of several emails from the tested domain. Moreover, it would also mandate a more complex infrastructure allowing the manipulation of, say, the X.509 certificate of our system. Therefore, it is presumed that if a domain endorses X.509, DANE, and MTA-STS in the inbound channel, it supports them in the outbound too.

B. MECSA

MECSA [9] has been developed with two goals in mind. First, to allow the ordinary end-user to obtain easy to parse and understand information about the security posture of their email provider when talking to other providers, and thus raise collective awareness on this subject. Second, to provide email system administrators with a platform where they can evaluate their email services, obtaining as a result a detailed report on the security of their MTA-to-MTA communications. A high-level view of MECSA along with the basic steps comprising the process of assessing the security level of the service under scrutiny is depicted in figure 2.

MECSA has been built with a privacy-by-design approach, thus minimising the amount of personal data exposed to the service. For service acquisition, the user must provide a valid email address. However, we only keep the domain name and delete the *username* part immediately after the completion of the tests mentioned in section III-A. The pair of email messages required for realising the evaluation service per every user request are also deleted right after the end of each assessment. The user IP address is also recorded in the logs for a limited number of days for security reasons.

As observed from figure 2, MECSA comprises four independent components; the web interface, the email server, the DB, and the report generator process. The figure also illustrates the different messages exchanged after a user submits a request for a new security evaluation.

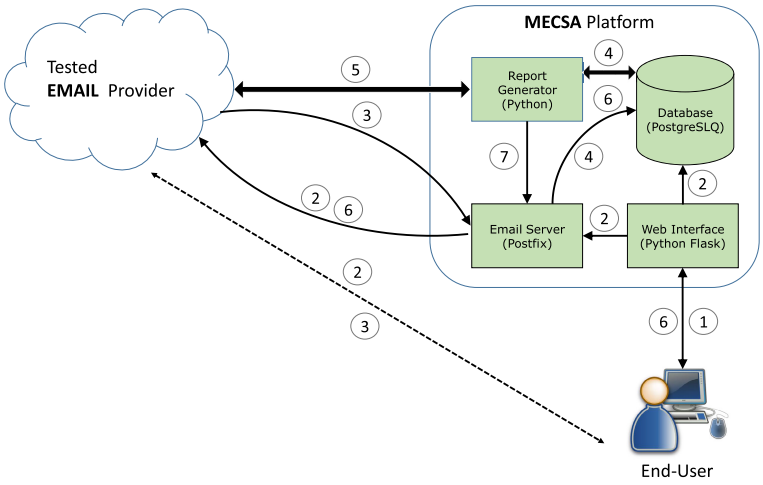

FIGURE 2. High-level view of the MECSA architecture. (1) The user voluntarily submits their email address, say, *user@domain.eu* to the service, (2) the request is registered, and an email is sent to the user, thus establishing an inbound connection with MECSA, (3) The user receives the email and replies to it, thus establishing an outbound connection with MECSA, (4) MECSA registers the messages and triggers the reporting process, (5) The report is generated based on (i) the outbound connection and the received message (outbound checks), and (ii) the information collected after attempting to establish TLS (STARTTLS) tunnels with all the MXs of the examined domain and querying the DNS for relevant RRs (inbound checks), (6) The relevant information, say, if a standard is supported or not are stored in the DB, (7) A unique report ID is communicated to the user who can enter it to the MECSA webpage interface to obtain the report.

The web interface has been developed in Python, using the Flask [40] framework. As it can be easily observed from the corresponding website [9], its main functions are to allow a user to submit a new request protected by a CAPTCHA, and to display the resulting reports, which are not public, but can be accessed using a unique *report_id* communicated to the user. For the sake of security, each request for a test generates a unique identifier, which is in turn included in the subject of the confirmation sent. Every time MECSA receives a reply, it checks if the subject and domain correspond to an entry in the DB.

The well-known open source MTA *Postfix* [41] was used as the email server. We employed a X.509 certificate from *Let's Encrypt* [42], and we configured DKIM and SPF to sign the messages sent and to validate the signatures as well as the sender of any received email. MECSA also supports MTA-STS, DANE, DMARC, and DNSSEC. However, for the sake of realising the tests, we employ a "weak" configuration, without enforcing the respective policies. That is, the objective is to be able to communicate to anyone, even in cases where the peer's parser is broken.

PostgreSQL [43] was used as the DB. The main reason for this choice over other alternatives like MySQL is the existence of the *LISTEN* and *NOTIFY* functions, which allow for monitoring the DB and waiting for a specific event. This functionality is used to trigger the generation of a report. For efficiency reasons, the DB also serves as a temporary cache with the aim of serving already cached reports before initiating new ones for the same user and email address. For the same purpose, the DB caches for up to 1 hour information on the SOA and DNSSEC domains, the revocation lists downloaded and the intermediate certificates.

The platform targets both non-technical and security-savvy users, which is a challenge. Reports focused on helping non-technical users to understand MTA-to-MTA communications security, may seem too simple for advanced users, and at the same time, reports targeting technical users may be too complicated for the general population. To address this problem, we have created a combined report and divided it into two parts: a summary report and an advanced report. A detailed description of these two types of reports along with the scoring system used are given in the MECSA website [9].

C. MECSA STANDALONE TOOL

MECSA-ST comprises an open source command line version of the MECSA service engine [10] under the EUPL 1.2 License [44]. This standalone tool is only fed with the domain name of the email service to be analysed as a command line argument, and hence it can be easily deployed to run large-scale unobtrusive automated tests involving an abundance of domains. For instance, the tool can take as input a list of domains obtained from the Google transparency report [45], the Majestic million [46], the Alexa top million websites that advertise mail servers [47], or the Cisco Umbrella 1 Million [48].

On the downside, MECSA-ST complete tests are limited to the analysis of the inbound email services as detailed in subsection III-A. Recall from the same subsection that

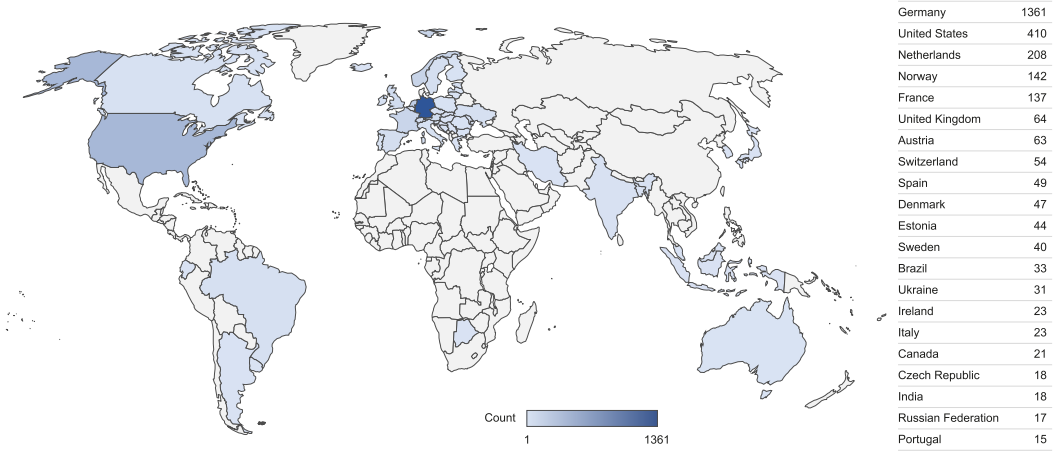

FIGURE 3. Evaluations per country based on the IP address of the received email. The right-side legend contains countries that accumulated at least 15 evaluations of unique email providers.

outbound channel analysis would require the reception of at least one email sent by the tested email provider. So, for DKIM, SPF, and DMARC, the tool provides only an estimate as follows.

- SPF and DMARC: The tool queries the DNS for SPF and DMARC DNS TXT RRs, and if the corresponding RR exists it is checked for syntax errors.
- DKIM: In this case, the corresponding DNS record cannot be located and fetched, because this would require to know beforehand the *selector* value as explained in subsection II-B. Nevertheless, an estimation can be made assuming that the NameServer (NS) of the assessed domain follows RFC 2308 [49]. Thus, the tool dispatches a DNS request to the NS of the domain, requesting the URL *_domainkey.somedomain*. If the NS follows the aforementioned standard, and the domain supports DKIM, the NS should respond with "NOERROR", otherwise it should return "NXDOMAIN".

D. DATASET

With the purpose of providing an up-to-date worldwide view of the adoption rate of the various email security standards outlined in subsection II-B, this work capitalises on the results collected by MECSA during a period of 15 months, i.e., from the 1st of Jan. 2019 to the 31st of Mar. 2020. As explained in subsection III-B, MECSA evaluations consider both the inbound and outbound channel, and thus, as further explained in section V, unlike most of the previous work in the literature, are full-fledged. In the above mentioned time span, MECSA conducted 3,236 evaluations of unique email providers (domains) scattered across 54 different countries. Note that the term "unique" refers also to the latest evaluation conducted per distinct email provider. About 47.5%, 29%, 10%, 4.5%, and 9% of these domains have one, two, three, four, or more than four MX RRs, respectively, while 2 domains presented no such RR. The geographic dispersal of unique email domains per country is given in figure 3. It is also to be noted that the dataset includes evaluations taken from top email providers, including hotmail.com, gmail.com, yahoo.com, aol.com, outlook.com, gmx.de, mail.ru, web.de, libero.it, yandex.com, zoho.com, protonmail.com, icloud.com, and several others.

E. LIMITATIONS

We consider two basic limitations. The first has to do with the corpus used. Specifically, about 80% and 15% of all the evaluations pertain to European and North America email domains, respectively. In this sense, while the dataset is quite balanced from a large versus small providers viewpoint, is rather skewed toward those that reside in the European and North American continents. Of course, this limitation is out of our control given that MECSA is a voluntary service.

The second pertains to the purposely non-intrusive nature of the service, meaning that the checks done are limited to those given in subsection III-A. For instance, MECSA does not check what the SMTP servers of the examined domain do when a policy, say, SPF is violated; will the server refuse the email by applying the policy or not? Instead, MECSA only examines whether the domain has a syntactically correct and proper SPF policy adverstised. On the other hand, the non-intrusive aspect is only to be loosely regarded as a hindrance, because the basic requirement is to deliver a public service that cannot be exploited by anyone to mount Denial of Service (DoS) attacks.

A last remark is that, as already pointed out in subsection II-A, the results presented in this paper pertain solely to the security of email communications between email providers. To be more precise, MECSA only assesses the implementation of the set of email security standards mentioned in subsection II-B. Therefore, this work does not address the security of the MTA itself, including password policy, firewall rules, software vulnerabilities, and so forth.

TABLE 1. Ciphersuite usage in the inbound channel. Ciphersuites with less than 4 appearances are omitted. R = Recommended, S = Secure, O = Obsolete.

Inbound ciphersuite		%	
TLSv1.2	ECDHE_RSA_AES_256_GCM_SHA384	63.86	R
TLSv1.2	ECDHE_RSA_CHACHA20_POLY1305	10.75	R
TLSv1.2	DHE_RSA_AES_256_GCM_SHA384	7.91	R
TLSv1.2	ECDHE_RSA_AES_128_GCM_SHA256	5.54	R
TLSv1.2	ECDHE_RSA_AES_256_CBC_SHA384	4.73	S
SSLv3	DHE_RSA_AES_256_SHA	1.73	O
TLSv1.2	ECDHE_ECDSA_AES_256_GCM_SHA384	1.39	R
TLSv1.2	DHE_RSA_CAMELLIA_256_CBC_SHA256	1.21	S
TLSv1.2	RSA_AES_256_GCM_SHA384	0.61	S
SSLv3	RSA_AES_128_SHA	0.58	O
TLSv1.0	ECDHE_RSA_AES_256_CBC_SHA	0.55	O
TLSv1.2	DHE_RSA_AES_256_CBC_SHA256	0.42	S
SSLv3	RSA_AES_256_SHA	0.21	O
TLSv1.2	ECDHE_ECDSA_CHACHA20_POLY1305	0.12	R
TLSv1.2	ECDHE_ECDSA_AES_128_GCM_SHA256	0.12	R

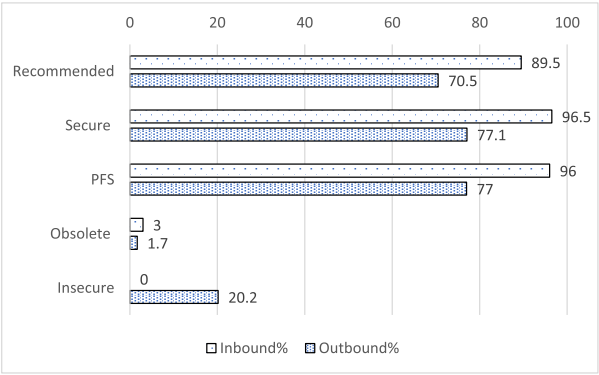

FIGURE 4. Cumulative (≈) results regarding the use of TLS ciphersuites.

And naturally, the security standards analysed comprise mitigation measures that aid in preventing attacks, they do not eradicate a threat.

IV. RESULTS

The current section presents our findings on the dataset with reference to subsection III-A. That is, we first detail on the results obtained for the inbound channel, followed by those obtained when inspecting the outbound stream.

A. INBOUND CHANNEL

Recall from section III-A that all checks in the inbound channel apply to all the MTAs per unique domain, as advertised in the corresponding DNS MX RRs, if any, or in the A type DNS RR if none MX RR is present.

1) STARTTLS

Regarding STARTTLS, the results revealed that 97.6% of the examined domains afforded at least one MTA with STARTTLS enabled. Favourably, this percent remained high even for domains for which 50%, 75%, and 100% of their MTAs, as given in the corresponding MX RRs, offer STARTTLS, i.e., 97.4%, 95.4%, and 94.3%, respectively. Also, for four domains, the establishment of a TLS session failed although the domain's MTA did advertise STARTTLS.

Table 1 summarises the percentage of usage of the different TLS protocol versions and ciphersuites in the inbound channel. Also, figure 4 depicts the most noteworthy pertinent results vis-à-vis those recorded for the outbound communication stream. Given that RFC 8446 [50] defines the AES_128 or AES_256 in GCM or CCM mode with SHA_256 or SHA_384, or the CHACHA20_POLY1305_SHA256 ciphersuites for use with TLS 1.3, it is concluded that ≈89.5% of the inbound MTAs in the dataset did apply a recommended ciphersuite, while an additional ≈7% a secure one. Indicatively, the work in [51], which is further considered in section V, reported a 51.5% usage of the TLSv1.2 ECDHE_RSA_AES_128_GCM_SHA256 ciphersuite for Gmail inbound traffic on Apr. 30, 2015. On the downside, it was observed that, still, ≈3% of the MTAs employed obsolete versions of the protocol, namely SSLv3 and TLSv.1.0. The results also suggested that an ≈86.5% of the receiving MTAs that employed a recommended or secure ciphersuite did favor Perfect Forward Secrecy (PFS) via the use of Elliptic-curve Diffie–Hellman Ephemeral (ECDHE) key exchange. An additional ≈9.5% offered the same security property by means of Diffie–Hellman Ephemeral (DHE) exchange, which however is much less efficient in terms of compute and network bandwidth resources than ECDHE.

2) PKIX CERTIFICATES

The analysis of the inbound channel during the TLS handshake phase, yielded 2,769 unique X.509 certificates. It is important to note that all the certificates pertaining to valid TLSA RRs have not been included in the aforementioned number. Further examination of the PKIX certificates showed that 72.3% of them satisfied all the checks enumerated in subsection III-A, namely, FQDN and all signatures were sound, the validity period was correct, and the certificate had not been revoked. On the adverse side, ≈22.3%, ≈17.3%, and ≈8.2%, of the certificates failed FQDN, signature, and expiration date validation, respectively. Interestingly, out of the certificates that failed CA signature validation, but passed the rest of the checks, 19% did not provide the intermediate certificates, i.e., the whole chain. No certificate was self-signed, and merely two were found to be revoked. Table 2 contains the top 10 certificate providers as seen in the inbound channel. As observed from the table, *Let's Encrypt* holds the largest share by far, while altogether these organisations account for ≈61% of the total number of certificates. Based on these figures, we can infer that the considerable number of self-signed certificates observed in MECSA 2018 evaluations have been now transitioned to Let's Encrypt ones, which is of course on the positive side.

Concerning the length of the RSA keys used in the certificates presented by the servers, we observed that the great mass of them (67.7%) utilised 2048-bit keys and another 27.5% 4096-bit or larger ones (0.14%). An ≈2.5% employed weak keys having a length less than 1024-bits. The work

TABLE 2. Top 10 certificate providers.

Certificate authority	Count
Let's Encrypt	898
DigiCert Inc.	267
COMODO CA Limited	176
GMO GlobalSign Ltd	103
Sectigo Ltd	64
TERENA	40
T-Systems International GmbH	38
Verein zur Foerderung eines Deutschen Forschungsnetzes e.V. (The German national research and education network)	36
Google Trust Services LLC	35

in [51], reported a 86.4% use of 2048-bit keys based on April 2015 data. Naturally, the improvement of ≈9% for keys equal or greater than 2048-bits is on the positive side, but not remarkable considering the five years distance.

3) DANE AND MTA-STS

With respect to DANE, nearly a fourth of the different domains, that is, ≈24.8%, had at least one TLSA RR. However, if we subtract invalid TLSAs, this percentage drops slightly to 24% or to 23.3% if we also take out those which finally failed certificate validation, say, due to FQDN mismatch or expiration date, or to ≈17.6% if we only count domains that afford DNSSEC-protected TLSA RRs. However, an overall 75.7% of the error-free DANE-enabled domains did protect the respective TLSA RRs by means of DNSSEC. Also on the plus side, an ≈84.8% of the domains with DANE support have published TLSA RR for all of their MTAs. Regarding TLSA certificate usage, 92.5% of the valid RRs specified the DANE-EE(3), and another ≈7% the DANE-TA(2) one. However, there were also a 0.5% that erroneously designated PKIX-EE(1). In closing, most of the valid TLSA RRs (≈96%) denoted a matching type of SHA2-256(1), while the rest that of SHA-512(2).

Only 5.6% of the receiving MTAs supported MTA-STS, all of them with a single valid TXT record. The policy mode for 69% of them was "enforce", while the rest 31% were configured with the "testing" mode. This is naturally an indication that several domains are in the progress of deploying this standard. Nevertheless, all in all, MTA-STS is by far the standard with less support. But this does not come as a surprise because the corresponding RFC [30] is the newest among all security standards included in subsection II-B.

4) DNSSEC

As discussed in subsection II-B, DNSSEC is tightly bound to all the standards of interest but MTA-STS. From the total number of the 3,236 distinct domains tested merely a 23.23% fully supported DNSECC with reference to the tests mentioned in subsection III-A. To ease the navigation through the various support rates per examined standard, we summarise them in figure 5. From the upper bars (first category) of the bar chart, which are computed over the total number of domains in the dataset, it is observed that the most proliferate standard is SPF followed by DKIM, DMARC, and DANE

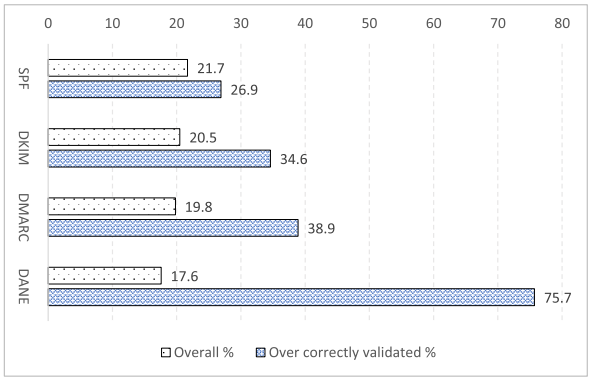

FIGURE 5. Overview of DNSSEC support rate per examined standard.

in that order. Overall, this score is poor as it lags behind by 11.6% of even the one-third of the domains in the best case. Section 5 also provides some justification about this disquieting result. On the other hand, with reference to the lower bars, which are computed over the number of domains that did support the respective standard, this situation is inverted. This latter outcome actually does not come as a surprise; for its security model to properly function, DANE requires the DNS RRs to be signed with DNSSEC.

B. OUTBOUND CHANNEL

Bear in mind that, differently to the inbound channel, the results presented in this subsection pertain to a single sending MTA per domain, that is, the one forwarding the user's reply mentioned in step ③ of figure 2. Nevertheless, a domain may afford more than one MTAs, and thus the following assessments accommodate all the different unique MTAs observed per a given domain.

1) STARTTLS

In the outbound channel, 96.4% of the sending MTAs successfully established a TLS session. The results regarding ciphersuite usage from an outbound MTA standpoint are gathered in table 3 and compared against those of the inbound channel in figure 4. It is concluded that ≈70.5% of the sending MTAs favored a ciphersuite included in RFC 8446, which is however significantly worse by a 19% than that observed for the inbound channel. This result can be primarily explained by the fact that our *Postfix* server is configured to accept any ciphersuite without any preference, meaning the first one offered by the client. Also, almost equal to the inbound channel, an additional ≈6.6% employed a currently secure ciphersuite. With regard to perfect forward secrecy, ≈77% supported this security property via the use of ECDHE or DHE key exchange. Nearly ≈1.7% of the sending servers employed an outdated version of the protocol, which is almost half of the corresponding percent observed for the inbound channel. Nevertheless, the most worrisome result is the preference for TLSv1.2 ciphersuites involving either anonymous ECDH (AECDH) or anonymous Diffie–Hellman (ADH) key

TABLE 3. Ciphersuite usage in the outbound channel. Ciphersuites with less than 5 appearances are omitted. R = Recommended, S = Secure, I = Insecure, O = Obsolete.

Outbound ciphersuite		%	
TLSv1.2	ECDHE_RSA_AES_256_GCM_SHA384	57.33	R
TLSv1.2	AECDH_AES_256_CBC_SHA	12.59	I
TLSv1.2	ECDHE_RSA_AES_128_GCM_SHA256	11.38	R
TLSv1.2	ADH_AES_256_GCM_SHA384	7.57	I
TLSv1.2	ECDHE_RSA_AES_256_CBC_SHA384	5.88	S
TLSv1.2	DHE_RSA_AES_256_GCM_SHA384	1.91	R
TLSv1.0	ECDHE_RSA_AES_256_SHA	0.45	O
TLSv1.2	ECDHE_RSA_AES_256_CBC_SHA	0.41	S
TLSv1.1	ECDHE_RSA_AES_256_CBC_SHA	0.41	O
TLSv1.0	ADH_AES_256_CBC_SHA	0.32	O
TLSv1.2	ECDHE_RSA_AES_128_CBC_SHA256	0.32	S
TLSv1.2	RSA_RC4_128_SHA	0.22	I
TLSv1.0	DHE_RSA_AES_256_SHA	0.16	O
TLSv1.0	AES_256_SHA	0.16	O
TLSv1.0	AECDH_AES256_SHA	0.16	O

exchange, reaching a total of ≈20.2%. Letting aside the fact that this category of ciphersuites do not use ephemeral keys, they are generally insecure as they are vulnerable to MITM attacks.

2) SPF

An approximately 80.7% of the tested domains' sending MTAs successfully passed SPF validation, i.e., the corresponding TXT DNS RR existed and it was syntactically correct. Nonetheless, this percentage drops to ≈26.9% if we reckon only the domains that additionally its SPF TXT RR is DNSSEC-protected. The latter percentage is even smaller, namely about 21.7%, if it is given as a ratio of the total number of the examined domains. As illustrated in the top three horizontal bars of figure 6, an approximately 97.8%, of the correctly validated emails was marked as "Pass" in the *Received-SPF* header of the received message. A 1.4%, 0.8% were characterised as "Softfail" or "Neutral", respectively. A further analysis on the syntactically valid SPF RRs pictured in the five bottom bars of the same figure, revealed that the majority, but still less than half, of the domains (44.12%) applied a *fail* policy. On the negative side, a 37.41% designated a *softfail* policy, meaning that the host should accept the mail, but label it as an SPF failure, and another 8.2% used a *neutral* policy, which as per RFC 7208 must be treated as identical to the *none* result. RFC 7208 clearly states that "It is better to use either the "redirect" modifier or the "all" mechanism to explicitly terminate processing". However, our results showed that 6.35% of the domains had published a policy with neither the "redirect" nor the "all" keywords. Even, a 0.12% applied a "+all" policy which is practically useless.

For the rest ≈19.3% of MTAs which failed SPF validation, the most frequent errors are summarized in table 4. The most common mistake was related to email forwarding, i.e., SPF validation will fail unless the sender's IP address in the original message is replaced by that of the forwarder. The rest of the errors are rather self-explanatory, e.g., the second

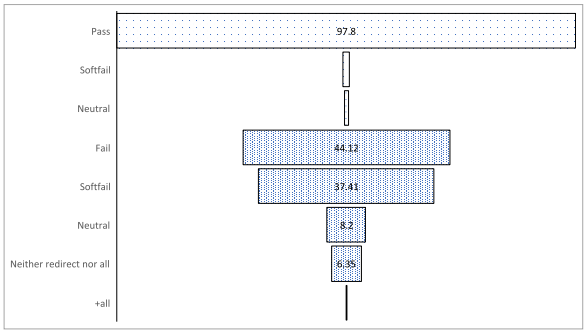

FIGURE 6. SPF validation percentages and SPF RRs (%).

error indicates a faulty authorisation, the seventh one notifies that the SPF RR could not be resolved within the maximum number of 10 DNS queries, while errors 3, 6, 8, and 9 are typically due to a syntax or format error in the SPF RR.

3) DKIM AND DMARC

Nearly 59.4% of the received emails had a *DKIM-Signature* header, about 21% less than that of SPF. At the bright side, signature validation failed for only ≈0.12% of them. Also, an analysis of the witnessed outbound MTAs of those domains in terms of diverse < *selector*>._*domainkey* values revealed that all afforded a corresponding DNS TXT RRs up to a percentage of 99%. No less important, only ≈34.6% of the DKIM-enabled domains (or about 20.5% of the total) protected their DKIM TXT RR by means of DNSSEC.

DMARC on the other hand, showed a 51.3% adoption rate among the examined providers. Only 0.31% of the aforementioned percentage pertained to syntactically invalid RRs, showing two types of errors: (a) "Error in Sanity Check DMARC record: Malformed email address in xx tag" or "Unknown URI found in xx tag" or "Invalid pair tag-value found", and (b) "Error parsing DMARC record: Duplicated tag found" or "Invalid pair tag-value found" or "Empty value found". Also, an analysis of the observed outbound MTAs of those domains revealed that all afforded a corresponding DNS TXT RRs up to a percentage of 99%. Regarding the policy announced by the valid DMARC TXT RRs, 42.1%, 33.2%, and 24.7% apply to the "none", "reject", and "quarantine" policies, respectively. An additional interesting observation is that the optional parameter "pct" mentioned in subsection II-B has been used in 22% of the policies; this ratio is almost equally distributed to a percentage of ≈10% for pct = 100 and pct = 10. No less important, DMARC is meant to be deployed along SPF and/or DKIM. Our results showed that ≈48.4% of the domains have deployed both DMARC and SPF, 44% both DMARC and DKIM, and 42.3% all the three standards. Lastly, in relation to DNSSEC, our results revealed that ≈38.9% of the domains which presented a valid DMARC TXR RR did support DNSSEC too. This percentage is translated to a 19.8% of the total number of domains.

TABLE 4. Observed SPF errors.

Reason	%
1. 250 SPF validation soft failure	86.78
2. 550 SPF fail - not authorized	2.38
3. 550 SPF Permanent Error: Two or more type TXT SPF records found	2.07
4. 451 SPF Temporary Error: DNS: TCP Fallback error: Conn. timed out	2.07
5. 550 SPF Permanent Error: Include has trivial recursion	1.75
6. 550 SPF Permanent Error: No valid SPF record for included domain	1.59
7. 550 SPF Permanent Error: Too many DNS lookups	1.11
8. 550 SPF Permanent Error: Invalid IP4/IPv6 address	0.79
9. 550 SPF Permanent Error: Unknown mechanism found	0.63

C. TAKEAWAYS

The current section recaps in brief the results given in the previous two subsections and rolls them up into takeaway points.

- As illustrated in figure 7, after considering the cumulative scores per examined standard over the whole period, i.e., from Jan. 2019 to Mar. 2020, we perceive an overall increase of 8.9%, 6.17%, 4.5%, 4.12%, 3.53%, 2.68%, and 2.21%, for DMARC, DKIM, SPF, DNSSEC, DANE, STARTTLS, and MTA-STS, respectively. Note that the first entries designating support of MTA-STS in MECSA's DB were logged on Jul. 2019 (0.43%). Also, the percentage given for STARTTLS accounts for email providers with at least 80% of its mail servers supporting this standard, counting both the inbound and outbound channels. In a nutshell, the above mentioned numbers bespeak a steady but not steep or major growth across all the standards over the 15 months period, with most promising that of DMARC. Naturally, the use of publicly available email security assessment services like that offered by MECSA has contributed to the increase of email security awareness, thus leading several providers to improve the security posture of their service.
- The support of STARTTLS reaches 97% as an average score for both channels. With reference to subsection IV-A1, this mean percentage diminishes by ≈1.7% if we only tally domains that all of their receiving MTAs support STARTTLS. This is a favourable result especially if seen in conjunction with the related work given in section V.
- The preference for the use of strong TLS ciphersuites in the inbound channel is in the vicinity of 90%, but falls short by 19% when it comes to the outbound one. The support for ciphersuites enabling perfect forward secrecy exhibits the same picture; 96% versus 77%. The use of obsolete versions of the protocol is low for both channels, but the employment of insecure ciphersuites up to a ≈20% in the outbound channel is an alarming issue.
- Regarding the PKIX certificate usage, the results indicate that there is still a considerable percentage of ≈26.7% of unwitting or not misconfigurations, with most common those that lead to a FQDN validation error. On the bright side, the certificates with the currently suggested [52] RSA 2048-bit or greater key sum up to more than 95%.
- Figure 5 outlines the level of DNSSEC support per affected standard. All in all, the ≈23% of full-fledged DNSSEC support by email providers in our corpus attests that (a) for many domains, the complexity of DNSSEC hinders its adoption, and as further explained in section V, its rollout down to Second-Level Domains (2LDs) is done at a rather glacial pace, and (b) due to the preceding point, the average support score for SPF, DKIM, DMARC, and DANE over the total numbers of domains examined is ≈20%.
- According to our results, SPF is by far the most mushroomed standard (≈81%), followed by DKIM (−21%), DMARC (−29%), and DANE (−57%) in that order, where the approximate deviation percentage is shown in parenthesis. Focusing on DMARC, its combination with either SPF or DKIM or both is unsatisfactory as does not embrace even half of the examined domains, i.e., in the best-case scenario, less than 49% for DMARC plus SPF. Despite that, given the low support rate of DNSSCEC, these numbers give a somewhat deceitful picture of email security vis-à-vis to actual DNSSEC-provided security. On the plus side, when referring to SPF- and DKIM-enabled domains, email validation in terms of the Received-SPF and DKIM-Signature headers succeeded to an average percentage of ≈99%. When it comes to DANE, it was really auspicious to see that more than the two-thirds of the DANE-enabled domains support DNSSEC, employing the correct type of certificates at the same time. In a nutshell, the above-mentioned figures suggest that so far, a considerable mass of email providers, along with the respective software providers, deploy the standards of interest correctly. In fact, as observed from the obtained results, e.g., more than 45% usage of "softfail" or "neutral" policies in SPF, and ≈42% usage of the "none" policy in DMARC, a significant number of email providers are transitioning toward these standards. Nevertheless, the overall progress can be characterised as rather moderate, and certainly feeble if it is contemplated in conjunction with the deployment rate of DNSSEC.

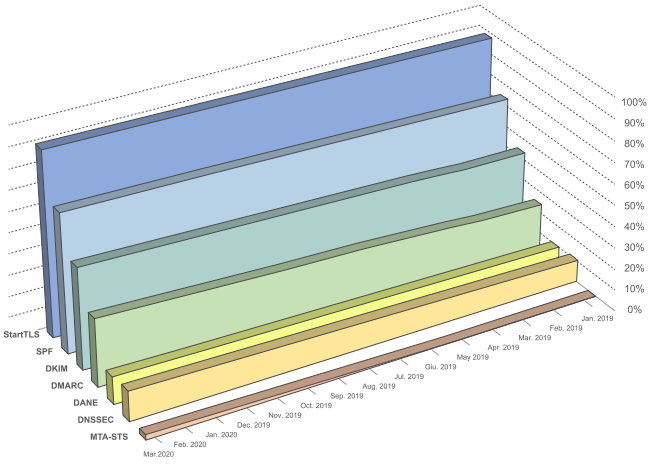

FIGURE 7. Historical adoption rate trend line per examined standard.

- On the grounds that the MTA-STS standard is quite new, our results indicate that it has been hitherto adopted by a tiny percentage of the email providers, i.e., less than 6%. The encouraging point, however, is that nearly 70% of the respective RRs were configured using the ''enforce'' mode. While MTA-STS does not depend on DNSSEC, it is vulnerable to downgrade attacks exercised by active attackers who can obstruct DNS responses when fetching an MTA-STS policy. Under this prism, MTA-STS may provide an illusory level of security.

V. RELATED WORK

Previous work in this area is rather scarce. To assist the reader's navigation through the various key results, including ours, we summarise them in table 5. In 2014, Facebook made a blog post reporting the use of STARTTLS when sending notification emails to their users. They concluded that while ≈60% of all the emails were sent over TLS, only ≈28% afforded strict certificate validation. All the rest, i.e., a 42% were transmitted in cleartext. During the same year, the authors in [53] measured the use of STARTTLS over SMTP as seen from the perspective of more than 100 Dutch organisations. Their results revealed that STARTTLS was supported or not supported by 55% and 34% of the examined domains, respectively. Also, for a 11% of the domains the test were inconclusive.

About one year latter, the authors in [51] presented an empirical analysis of STARTTLS, SPF, DMARC and DKIM based on two datasets, namely Google transparency report (Gmail logs from Jan. 2014 to April 2015) and the April 2015 Alexa list of top 1M ranked websites. Their work included a study on DNS Hijacking, but neither DANE nor DNSSEC usage by SMTP servers were considered. As a side contribution, the authors present an interesting study on STARTTLS corruption, i.e., TLS stripping attacks, from an active attacker's viewpoint. For the Gmail corpus, the authors observed a peak score of 60% and 80% of all the inbound and outbound email sent over TLS, respectively. However, they pinpoint that this result was biased by major email providers, and indeed this inference was verified after examining a sample of ≈900 most common domains interacted with Gmail, where only 58% and 29% of them accepted or transmitted encrypted outbound and inbound mail respectively. Regarding the email-enabled domains stemming from the Alexa dataset, they concluded that ≈82% supported STARTTLS. For SPF, DKIM, and DMARC the authors reported a best result of 92%, 83%, ≈26%, respectively for the Google dataset. For the Alexa one, they observed a 47% and ≈1% for SPF and DMARC, respectively. Nevertheless, even for the domains that did support these standards, several errors or misconfigurations were spotted, including faulty policies, weak or revoked keys, and protocol hiccups.

During about the same period, i.e., March 2014 to Feb. 2015, the authors in [54] conducted a major measurement study on the adoption of some of the security technologies of interest. Their study was not merely focused on provider-to-provider communications, but also considers POP and IMAP protocols. They relied on two datasets, one comprised of ≈300K major email providers, and a much larger incorporating popular providers existed in the Alexa top 1M and the leaked Adobe user dataset of Sept. 2013 [55]. Regarding the obtained results, the authors reported a 89%, 85%, 68% support for STARTTLS, SPF, and DMARC, respectively for incoming connections in the Adobe corpus. These results however are weighted by the number of users for STARTLS, and by the frequency of occurrence for SPF and DMARC. The authors also studied the level of DNSSEC adoption by major providers in the Adobe and Alexa lists, and concluded that ≈3% have enabled this security extension. Another interesting observation is that 13% and 23% of the DNSSEC-enabled hosts in the Alexa and Adobe lists respectively were improperly configured.

As observed from table 5, for SPF, the various results are quite close except the one pertaining to the Alexa dataset in [51], which however seems more natural for the time that research was conducted. Regarding DKIM, the high percentage reported by [51] is probably due to the skew toward major email providers as noted by the authors in the Gmail dataset. Lastly, with reference to our findings, the DMARC score has almost doubled if compared against that in [51]. This increment seems reasonable if we consider the time span between these contributions. The same conclusion can be inferred for the DNSSEC scores reported by [54] and our work.

By referring to the Google's transparency report for ''email encryption in transit'' [45], it can be calculated that the average per annum usage (%) of TLS from 2015 to 2019 for the inbound and outbound channel is correspondingly (57, 80), (74, 85), (86, 88), (90, 89), and (93, 90). These results coincide with those reported by [51], [54], and additionally reveal an overall 63% and 13% improvement for the inbound and outbound channel, respectively from 2015 up to 2019. Our results show a betterment of 4.6% and 6.4%, respectively, thus verifying this augmentation tendency.

TABLE 5. Summary of key results. All numbers are expressed as percentages (%).

Protocol	[61] 2014	[53] 2014	[51] 2015	[54] 2015	[45] 2018, 2019	MECSA 2019-20
STARTTLS	60/28[a]	55/34[b]	60/80/82[c]	89	90/89, 93/90[g]	97.6/96.4[g]
SPF	-	-	92/47[d]	85	-	80.7/26.9[h]
DKIM	-	-	83[e]	-	-	59.4/26.9[h]
DMARC	-	-	26/1[d]	68[f]	-	51.3/38.9[h]
DNSSEC	-	-	-	3	-	23.23
DANE	-	-	-	-	-	17.6
MTA-STS	-	-	-	-	-	5.6

[a] Supported / Supported with strict certificate validation. [b] Supported / Unsupported. [c] Supported Gmail inbound/Gmail outbound/Alexa [d] Gmail/Alexa. [e] Gmail. [f] According to the authors, this significant difference vis-à-vis [51] is due to the mix of providers in the datasets. [g] Inbound / Outbound. [h] Total / DNSSEC-enabled.

The authors in [56] conducted a systematic study of DANE TLSA deployment. As of Dec. 2014, they concluded that out of the 485k DNSSEC-secured .com and .net zones they monitored, only 997 TLSA names were spotted (≈0.2%). In addition, according to their observations (a) the deployment of DANE TLSA was consistently augmenting, and (b) about 7 to 13% of TLSA records were faulty. Regarding the former observation, our results show a far much better situation, with 17.6% of the examined domains to afford both DNSSEC and correctly delineated TLSA RRs. An analogous amelioration is noticed for the latter point, where only an 1.5% of faulty TLSA RRs was observed.

With respect to DNSSEC, the authors in [57] presented a comprehensive study on the deployment of DNSSEC over a period of 21 months, i.e., from March 2015 to Dec. 2016. According to their estimates, almost a third of DNSSEC-enabled domains were improperly configured, thus offering an illusion of security. This result is close to that of the authors in [54], stating that about one fourth of the DNSSEC-enabled hosts in the 2013 Adobe list were misconfigured. They also pinpointed that despite the fact that 83% of the observed resolvers did query for DNSSEC RRs, only 12% of them actually validated the returned records. The work in [58] reported that as of Jan. 2017, about 90% of TLDs have deployed DNSSEC. Recent statistics regarding DNSSEC deployment from [59] indicate that while 90% of Top-Level Domains (TLDs) are signed (note that this percentage is equal to that exhibited by [58]), only ≈4% of their Second-Level Domains (2LDs) are signed as well, and ≈24% of the end-users validate the returned RRs. Put simply, this rather feeble result means that hitherto only a small portion of end-users benefit from the deployment of DNSSEC.

Lastly, analogous to MECSA-ST (see subsection III-C) email security assessment tools are provided by (a) the *Internet.nl*, which is an initiative of the Dutch Internet Standards Platform [60], and (b) the *Webcheck.pt* which is a joint initiative of the Portuguese National Cybersecurity Center and the DNS.PT association. Upon entering the domain part of an email address, these tools test if the targeted email service offers support for DNSSEC, DMARC, DKIM, SPF, STARTTLS, and DANE (currently only for the former). At present, MTA-STS is not addressed by either of the aforementioned tools. As with MECSA-ST, all the aforementioned evaluations apply solely to the inbound channel as detailed in subsection III-C.

VI. CONCLUSION AND FUTURE WORK

The adoption of modern email security standards for MTA-to-MTA communications is hitherto nonobligatory. Consequently, an email provider can achieve a fair degree of interoperability with other providers without necessarily supporting state-of-the-art security standards and best practices [62]. For the interested reader, a subtle analysis of this interoperability versus security trade-off is given in [54]. Even worse, this lack of clear incentives spurring the adoption of strong security measures unavoidably propagates along the supply chain. In fact, this assertion was verified by the authors in [51]. From an end-user's viewpoint, this situation is doubly problematic because neither they have the means of knowing if and up to what point such standards are implemented by their provider under the hood nor their email communications are adequately protected in every possible network hop. They even suffer from, say, surges of unsolicited bulk email due to that very same shortcoming.

This work serves a dual purpose. First off, we develop an email evaluation platform that comes in two flavors; MECSA can be of help to both the ordinary user and email system administrators, while MECSA-ST might be of interest to researchers working in this area. The crux of the matter here is that for spurring improvements, in this case in the security status quo of email providers, one should be able to straightforwardly and massively assess it. Secondly, we capitalise on the plentiful and diverse data collected by our platform to conduct a large-scale timely assessment of the state-of-play in this area, focusing on seven different security standards. Vis-à-vis the related work, the outcomes of this endeavor are far more complete, given that they pertain to both communication streams and additionally consider newer standards or other aspects left unaddressed by prior work in this area. The results obtained are dual-faceted. On the one hand, they reveal significant shortages or hiccups in the rollout of some of these standards, with DNSSEC and MTA-STS to stand out due to different reasons. DNSSEC directly affects all other standards but MTA-STS, and hence is a key factor towards the

provision of actual security in the email ecosystem. On the other, we also witnessed favorable results, especially for STARTTLS, including the use of strong ciphersuites.

We are currently working on delivering further enhancements to the MECSA platform, including the support for the SMTP TLS Reporting [63] and RequireTLS [64] emerging standards.

REFERENCES

[1] The Radicati Group. (Feb. 2020). *Email Statistics Report, 2020-2024—Executive Summary*. [Online]. Available: https://www.radicati.com/wp/wp-content/uploads/2019/12/Email-Statistics-Report-2020-2024-Executive-Summary.pdf

[2] European Union. (Apr. 2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons With Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation)*. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/2016-05-04

[3] J. Klensin, *Simple Mail Transfer Protocol*, document RFC 5321, RFCs 7504, Draft Standard, Internet Engineering Task Force, Oct. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5321.txt

[4] A. Whitten and J. D. Tygar, "Why Johnny can't encrypt: A usability evaluation of PGP 5.0," in *Proc. 8th Conf. USENIX Secur. Symp.* Berkeley, CA, USA: USENIX Association, vol. 8, 1999, p. 14.

[5] K. Moore and C. Newman, *Cleartext Considered Obsolete: Use of Transport Layer Security (TLS) for Email Submission and Access*, doument RFC 8314, Jan. 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8314.txt

[6] H. Finney, L. Donnerhacke, J. Callas, R. L. Thayer, and D. Shaw, *OpenPGP Message Format*, document RFC 4880, Nov. 2007. [Online]. Available: https://rfc-editor.org/rfc/rfc4880.txt

[7] J. Schaad, B. C. Ramsdell, and S. Turner, *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification*," document RFC 8551, Apr. 2019. [Online]. Available: https://rfc-editor.org/rfc/rfc8551.txt

[8] J. Müller, M. Brinkmann, D. Poddebniak, H. Böck, S. Schinzel, J. Somorovsky, and J. Schwenk, "Johnny, you are fired!'–spoofing OpenPGP and S/MIME signatures in Emails," in *Proc. 28th USENIX Security Symp.* Santa Clara, CA, USA: USENIX Association, Aug. 2019, pp. 1011–1028. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/muller

[9] European Commission. (2020). *My Email Communications Security Assessment (MECSA)*. Accessed: Apr. 1, 2020. [Online]. Available: https://mecsa.jrc.ec.europa.eu/

[10] European Commission. (2020). *MECSA Standalone Tool*. Accessed: Apr. 1, 2020. [Online]. Available: https://github.com/mecsa/mecsa-st

[11] N. Freed and N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, document RFC 2045, RFCs 2184, 2231, 5335, 6532, Draft Standard, Internet Engineering Task Force, Nov. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc2045.txt

[12] N. Freed and N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, document RFC 2046, RFCs 2646, 3798, 5147, 6657, Draft Standard, Internet Engineering Task Force, Nov. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc2046.txt

[13] K. Moore, *MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text*, document RFC 2047, RFCs 2184, 2231, Draft Standard, Internet Engineering Task Force, Nov. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc2047.txt

[14] N. Freed, J. Klensin, and J. Postel, *Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures*, document RFC 2048, RFCs 4288, 4289, RFC 3023, Best Current Practice, Internet Engineering Task Force, Nov. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc2048.txt

[15] N. Freed and N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples*, document RFC 2049, Draft Standard, Internet Engineering Task Force, Nov. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc2049.txt

[16] P. Hoffman, *SMTP Service Extension for Secure SMTP over Transport Layer Security*, document RFC 3207, RFC 7817, Proposed Standard, Internet Engineering Task Force, Feb. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3207.txt

[17] J. Myers and M. Rose, *Post Office Protocol—Version 3*, document RFC 1939, RFCs 1957, 2449, 6186, Internet Standard, Internet Engineering Task Force, May 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1939.txt

[18] M. Crispin, *Internet Message Access Protocol—Version 4Rev1*, document RFC 3501, RFCs 4466, 4469, 4551, 5032, 5182, 5738, 6186, 6858, 7817, Proposed Standard, Internet Engineering Task Force, Mar. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3501.txt

[19] A. Melnikov, *Updated Transport Layer Security (TLS) Server Identity Check Procedure for Email-Related Protocols*, document RFC 7817, Proposed Standard, Internet Engineering Task Force, Mar. 2016. [Online]. Available: http://www.ietf.org/rfc/rfc7817.txt

[20] S. Kitterman, *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1*, document RFC 7208, 7372, Proposed Standard, Internet Engineering Task Force, Apr. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7208.txt

[21] D. Crocker, T. Hansen, and M. Kucherawy, *DomainKeys Identified Mail (DKIM) Signatures*, document RFC 6376, Internet Standard, Internet Engineering Task Force, Sep. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6376.txt

[22] M. Kucherawy and E. Zwicky, *Domain-based Message Authentication, Reporting, and Conformance (DMARC)*, document RFC 7489, Informational, Internet Engineering Task Force, Mar. 2015. [Online]. Available: http://www.ietf.org/rfc/rfc7489.txt

[23] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, *DNS Security Introduction and Requirements*, document RFC 4033, RFCs 6014, 6840, Proposed Standard, Internet Engineering Task Force, Mar. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4033.txt

[24] P. Hoffman, *Cryptographic Algorithm Identifier Allocation for DNSSEC*, document RFC 6014, Proposed Standard, Internet Engineering Task Force, Nov. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc6014.txt

[25] S. Weiler and D. Blacka, *Clarifications and Implementation Notes for DNS Security (DNSSEC)*, document RFC 6840, Proposed Standard, Internet Engineering Task Force, Feb. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc6840.txt

[26] P. Hoffman and J. Schlyter, *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA*, document RFC 6698, RFCs 7218, 7671, Proposed Standard, Internet Engineering Task Force, Aug. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6698.txt

[27] O. Gudmundsson, *Adding Acronyms to Simplify Conversations about DNS-Based Authentication of Named Entities (DANE)*, document RFC 7218, Proposed Standard, Internet Engineering Task Force, Apr. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7218.txt

[28] V. Dukhovni and W. Hardaker, *The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance*, document RFC 7671, Proposed Standard, Internet Engineering Task Force, Oct. 2015. [Online]. Available: http://www.ietf.org/rfc/rfc7671.txt

[29] V. Dukhovni and W. Hardaker, *SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS)*, document RFC 7672, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7672.txt

[30] D. Margolis, M. Risher, B. Ramakrishnan, A. Brotman, and J. Jones, *SMTP MTA Strict Transport Security (MTA-STS)*, document RFC 8461, Sep. 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8461.txt

[31] N. L. M. van Adrichem, N. Blenn, A. R. Lúa, X. Wang, M. Wasif, F. Fatturrahman, and F. A. Kuipers, "A measurement study of DNSSEC misconfigurations," *Secur. Informat.*, vol. 4, no. 1, pp. 1–14, Dec. 2015.

[32] H. Shulman and M. Waidner, "One key to sign them all considered vulnerable: Evaluation of DNSSEC in the Internet," in *Proc. 14th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*. Boston, MA, USA: USENIX Association, Mar. 2017, pp. 131–144. [Online]. Available: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/shulman

[33] T. Chung, R. van Rijswijk-Deij, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "Understanding the role of registrars in DNSSEC deployment," in *Proc. Internet Meas. Conf.* New York, NY, USA: Association Computing Machinery, Nov. 2017, pp. 369–383, doi: 10.1145/3131365.3131373.

[34] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, and S. Gritzalis, "DNS amplification attack revisited," *Comput. Secur.*, vol. 39, p. 475–485, Nov. 2013, doi: 10.1016/j.cose.2013.10.001.

[35] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, "Detecting DNS amplification attacks," in *Critical Information Infrastructures Security*, J. Lopez and B. M. Hämmerli, Eds. Berlin, Germany: Springer, 2008, pp. 185–196.

[36] P. Saint-Andre and J. Hodges, *Representation and Verification of Domain-Based Application Service Identity Within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)*, document RFC 6125, Mar. 2011. [Online]. Available: https://rfc-editor.org/rfc/rfc6125.txt
[37] S. Boeyen, S. Santesson, T. Polk, R. Housley, S. Farrell, and D. Cooper, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, document RFC 5280, May 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5280.txt
[38] Python Software Foundation. (2020). *Python Module PYSPF 2.0.14*. Accessed: May 11, 2020. [Online]. Available: https://pypi.org/project/pyspf/
[39] Python Software Foundation. (2020). *Python Module DKIMPY 1.0.4*. Accessed: May 11, 2020. [Online]. Available: https://pypi.org/project/dkimpy/
[40] Pallets. (2020). *Flask Micro Web Framework Project*. Accessed: Apr. 1, 2020. [Online]. Available: https://flask.palletsprojects.com/en/1.1.x/
[41] Postfix. (2020). *The Postfix Home Page*. Accessed: Apr. 10, 2020. [Online]. Available: http://www.postfix.org/
[42] ISRG. (2020). *Let's Encrypt Free, Automated, and Open Certificate Authority*. Accessed: Apr. 1, 2020. [Online]. Available: https://letsencrypt.org/
[43] The PostgreSQL Global Development Group. (2020). *PostgreSQL*. Accessed: Apr. 10, 2020. [Online]. Available: https://www.postgresql.org/
[44] European Commission. (2020). *European Union Public Licence*. Accessed: Apr. 1, 2020. [Online]. Available: https://ec.europa.eu/info/european-union-public-licence_en
[45] G. LLC. (2020). *Google Transparency Report–Email Encryption in Transit*. Accessed: Apr. 1, 2020. [Online]. Available: https://transparencyreport.google.com/safer-email/overview?hl=en
[46] Majestic. (2020). *The Majestic Million*. Accessed: Apr. 1, 2020. [Online]. Available: https://majestic.com/reports/majestic-million
[47] Alexa Internet. (2020). *Top Sites on the Web*. Accessed: Apr. 1, 2020. [Online]. Available: https://www.alexa.com/topsites
[48] Cisco Umbrella. (2020). *Cisco Umbrella 1 Million*. Accessed: Apr. 1, 2020. [Online]. Available: https://umbrella.cisco.com/blog/cisco-umbrella-1-million
[49] M. Andrews, *Negative Caching of DNS Queries (DNS NCACHE)*, document RFC 2308, Mar. 1998. [Online]. Available: https://rfc-editor.org/rfc/rfc2308.txt
[50] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, documemt RFC 8446, Aug. 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8446.txt
[51] Z. Durumeric, J. A. Halderman, D. Adrian, A. Mirian, J. Kasten, E. Bursztein, N. Lidzborski, K. Thomas, V. Eranti, and M. Bailey, "Neither snow nor rain nor MITM. . .: An empirical analysis of email delivery security," in *Proc. ACM Conf. Internet Meas. Conf. (IMC)*. New York, NY, USA: Association Computing Machinery, 2015, pp. 27–39, doi: 10.1145/2815675.2815695.
[52] E. Barker and A. Roginsky, "NIST special publication 800-131a rev. 2-sp 800-57. Transitioning the use of cryptographic algorithms and key lengths," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. SP 800-131A Rev. 2, 2019.
[53] S. Rijs and M. van der Meer. (2014). *The State of StartTLS*. Accessed: Apr. 1, 2020. [Online]. Available: https://www.os3.nl/_media/2013-2014/courses/ot/magiel_sean2.pdf
[54] 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 *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*. New York, NY, USA: Association Computing Machinery, 2015, pp. 450–464, doi: 10.1145/2810103.2813607.
[55] B. Arkin. (2013). *Adobe Important Customer Security Announcement*. Accessed: Apr. 1, 2020. [Online]. Available: https://theblog.adobe.com/important-customer-security-announcement/
[56] L. Zhu, D. Wessels, A. Mankin, and J. Heidemann, "Measuring dane TLSA deployment," in *Traffic Monitoring and Analysis*, M. Steiner, P. Barlet-Ros, and O. Bonaventure, Eds. Cham, Switzerland: Springer, 2015, pp. 219–232.
[57] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "A longitudinal, end-to-end view of the DNSSEC ecosystem," in *Proc. 26th USENIX Secur. Symp. (USENIX Secur.)* Vancouver, BC, Canada: USENIX Association, 2017, pp. 1307–1322. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/chung
[58] M. Wander, "Measurement survey of server-side DNSSEC adoption," in *Proc. Netw. Traffic Meas. Anal. Conf. (TMA)*, Jun. 2017, pp. 1–9.
[59] R. Lamb. (2020). *DNSSEC Deployment Report*. Accessed: Apr. 1 2020. [Online]. Available: http://rick.eng.br/dnssecstat/
[60] D. I. S. Platform. (2015). *Dutch Internet Standards Platform-Internet*. Accessed: Apr. 10, 2020. [Online]. Available: https://en.internet.nl/
[61] Facebook. (2014). Accessed: Apr. 1, 2020. *The Current State of SMTP STARTTLS Deployment*. [Online]. Available: https://www.facebook.com/notes/protect-the-graph/the-current-state-of-smtp-starttls-deployment/1453015901605223/
[62] A. Malatras, I. Coisel, and I. Sanchez, "Technical recommendations for improving security of email communications," in *Proc. 39th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2016, pp. 1381–1386.
[63] D. Margolis, A. Brotman, B. Ramakrishnan, J. Jones, and M. Risher, *SMTP TLS Reporting*, document RFC 8460, Sep. 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8460.txt
[64] J. Fenton, *SMTP Require TLS Option*, document RFC 8689, Nov. 2019. [Online]. Available: https://rfc-editor.org/rfc/rfc8689.txt

GEORGIOS KAMBOURAKIS has served as the Head of the Department, from September 2019 to October 2019. He was the Director of Info-Sec-Lab, from September 2014 to December 2018. He is currently a Full Professor with the Department of Information and Communication Systems Engineering, University of the Aegean, Greece. He is also on unpaid leave from the University, while he is working for the European Commission at the European Joint Research Centre (JRC), Ispra, Italy. His research interests are in the fields of mobile and wireless networks security and privacy, VoIP security, IoT security and privacy, DNS security, and security education. He has more than 135 refereed publications in the aforementioned areas. More info at: http://www.icsd.aegean.gr/gkamb

GERARD DRAPER GIL (Member, IEEE) received the Ph.D. degree in computer science from the University of the Balearic Islands, Spain, in 2013. He is currently a Scientific Project Officer with the Joint Research Centre of the European Commission. He previously worked as a Postdoctoral Fellow at the Information Security Centre of Excellence in Canada, and the University of the Balearic Islands. His main research interests are security and privacy of network protocols and services, digital forensics, malware analysis, network traffic analysis, and machine learning applied to cybersecurity.

IGNACIO SANCHEZ (Member, IEEE) received the M.Sc. degree in computer engineering from the University of Deusto, Spain, and the Ph.D. degree in computer engineering from the National Distance Education University (UNED), Spain. He holds the CISSP certification and has over 18 years of experience in the domain of information security. He is currently a European Commission official with the Joint Research Centre. He works in the Cyber and Digital Citizen Security Unit, within the Directorate on Space, Security and Migration, where he leads several lines of research in the fields of cybersecurity, privacy, data protection, and fight against cybercrime.