“Objection, Your Honor!”: False Positive Detection in Sender Domain Authentication by Utilizing the DMARC Reports

Kanako Konno
Amazon Web Services Japan K.K.
Tokyo, 141-0021, Japan
Email: kankon@amazon.co.jp

Naoya Kitagawa
Research and Development Center
for Academic Networks,
National Institute of Informatics
Tokyo, 101-8430, Japan
Email: kitagawa@nii.ac.jp

Nariyoshi Yamai
Division of Advanced Information Technology
and Computer Science Institute of Engineering,
Tokyo University of
Agriculture and Technology
Tokyo, 184-8588, Japan
Email: nyamai@cc.tuat.ac.jp

Abstract—**Information leakage and phishing scams caused by spoofed e-mails have become serious problems, particularly in the fields of business and e-commerce. Sender domain authentications, such as Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM), and Domain-based Message Authentication, Reporting, and Conformance (DMARC), are effective countermeasures against spoofed e-mails. In particular, DMARC is one of the most effective methods of sender domain authentication. However, sender domain authentication methods erroneously classify legitimate e-mails, such as forwarded e-mails, as malicious e-mails. Because sender domain authentication is usually processed prior to content filtering, the fact that sender domain authentications generate a large number of false positives is a serious problem. In this paper, we propose a method to detect false positive deliveries in sender domain authentications based on the legitimacy of the senders' IP addresses by adapting X-means clustering to the reports generated by the reporting function of DMARC. Our approach consists of three phases: DMARC report summarization, X-means clustering, and legitimate sender detection. Applied to actual DMARC reports, we found that our method detected 214,153 e-mails on average sent from 347 legitimate senders' IP addresses on average as legitimate e-mails per day. We evaluate our results focusing on the legitimate deliveries sent from the detected legitimate senders and the detected false positives generated by existing sender domain authentications. The evaluation results confirmed that our method can detect large numbers of legitimate e-mails, including the false positive e-mails, such as forwarded e-mails, which cannot be correctly identified using existing sender domain authentication technologies.**

Keywords–Spoofed e-mail; SPF; DKIM; DMARC; Clustering.

I. INTRODUCTION

This paper is an extended version of our previous study presented at the Eleventh International Conference on Evolving Internet [1]. In our previous study, we proposed a mechanism to detect e-mail forwarding servers, which are a type of legitimate e-mail sending server, via clustering. In this paper, as an enhancement to our previous study, we propose a method to detect many types of legitimate e-mail sending servers, in addition to forwarding servers. By utilizing our method proposed in this paper, e-mail system administrators can detect a variety of legitimate deliveries that have been false positives with conventional sender domain authentications.

E-mail is one of the most utilized communication services worldwide. However, especially in business, e-mail has a serious problem due to the rapid increase in information leakage and phishing scams enabled by spoofing e-mail. According to the statistics report of the Federal Bureau of Investigation, the total financial damage due to spoofed e-mails was 26.2 billion US dollars from June 2016 to July 2019 [2]. Spoofed e-mails are used by spammers to steal sensitive information or send malicious programs, such as computer viruses.

Sender domain authentication has been proposed as an effective countermeasure to spoofed e-mails. Sender Policy Framework (SPF) [3] and DomainKeys Identified Mail (DKIM) [4] are methods that are widely used. SPF is a method, in which the receiver confirms whether the e-mail sender's IP address is legitimate by checking the original sender's SPF record, which is a list of IP addresses that the sender may use to send e-mails. However, SPF cannot verify forwarded e-mails correctly because the sender's IP address is changed to the forwarder's IP address, which is not included in the sender's SPF record when the e-mails are forwarded. In DKIM, the receivers verify the digital signatures generated from the header and body of the e-mail and confirm whether the e-mail has been rewritten by spammers. DKIM allows a third-party's domain to sign e-mails; therefore, DKIM has the problem that spoofed e-mails signed by a spammer's own malicious domain will incorrectly pass its verification.

Domain-based Message Authentication, Reporting, and Conformance (DMARC) [5] is one of the most effective sender domain authentication frameworks and includes reporting and policy controlling mechanisms. DMARC utilizes both the SPF and DKIM authentication mechanisms to verify e-mails. DMARC has a reporting function that enables an e-mail sender to receive a “DMARC aggregate report” (hereafter, called the DMARC report). This report provides information, such as the header of the e-mail and the authentication results. In general, DMARC reports are used to confirm the effectiveness of sender domain authentications by e-mail senders. However, we can also observe the transmission behavior for each e-mail sending server by analyzing the information in the DMARC reports.

Anti-spam methods are generally operated in three phases: Transmission Control Protocol (TCP) and Simple Mail Transfer Protocol (SMTP) session monitoring and blacklist, sender domain authentication, and content filtering. In such an anti-

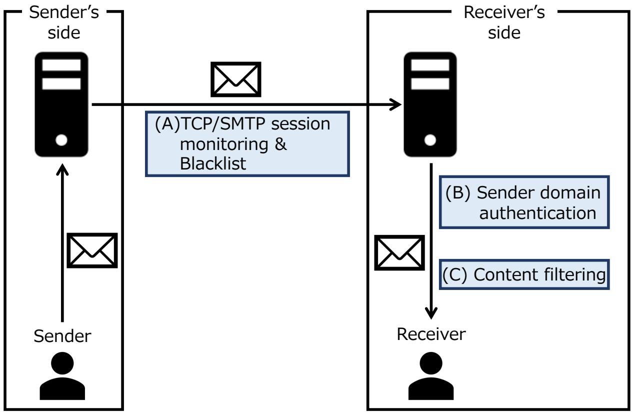

Sender's side
Receiver's side
(A)TCP/SMTP session monitoring & Blacklist
(B) Sender domain authentication
(C) Content filtering
Sender
Receiver

Figure 1. General flow of anti-spam measures.

spam operation, e-mail servers process sender domain authentications before implementing the content filtering method. Therefore, it is essential to reduce the number of false positives in the sender domain authentication. Conversely, the increase in false negatives in the sender domain authentication caused by reducing the number of false positives is not a critical problem.

In this paper, we propose a method to detect legitimate e-mail senders in order to reduce false positives in the sender domain authentication via X-means clustering [6] using the massive amounts of available DMARC report data. Our approach consists of three phases: DMARC report summarization, X-means clustering, and legitimate sender detection.

To test our approach, we apply it to actual DMARC report data. To evaluate our results, we investigate the details of the detected legitimate e-mails sent from the legitimate senders and the false positive deliveries in the sender domain authentications. Our evaluation results indicate that our method detects false positive e-mails, such as forwarded e-mails, which cannot be correctly determined by existing sender domain authentication technologies.

This paper organized as follows. In Section II, we explain several existing anti-spam methods. In Section III, we describe the design of our approach. Then, we describe the dataset that we use in our experiment in Section IV. Section V shows the results generated when applying our method to the dataset. In Section VI, we evaluate our results focusing on the number and ratio of false positive deliveries of the different sender domain authentication technologies. Finally, we present our concluding remarks in Section VII.

II. Related Work

Anti-spam measures are generally processed in phases. In this section, we show several approaches for each of the phases shown in Figure 1.

A. *TCP/SMTP session monitoring and blacklists*

Greylisting [7] is a method that checks the retry function for establishing an SMTP session. In general, legitimate e-mail senders try to resend an e-mail after a period of time when an e-mail is temporally rejected. Conversely, spammers who use massive e-mail sending tools do not try to resend e-mails. This technique, which takes advantage of such differences in sending behavior, is effective as a countermeasure against spammers sending large amounts of e-mail.

SMTP tarpitting [8] detects spam e-mails by delaying a response to the sender's server. Spammers generally try to send as many spam e-mails as possible in a short period of time. Therefore, they tend to ignore a response from a receiver's server or abandon sending the spam e-mails altogether. Even though SMTP tarpitting can eliminate such transmissions with priority on delivery efficiency, it also delays transmissions by legitimate senders. Therefore, legitimate e-mails may not be delivered correctly.

Kitagawa et al.'s method [9] inspects the SYN packet retry function for establishing a TCP session between a sending host and a receiving host. This method is effective for spam delivery that gives priority to e-mail delivery efficiencies such as greylisting and SMTP tarpitting.

Even though these methods are highly effective against conventional spam transmission, it is expected that the reduction effect for the cleverly spoofed e-mails that have become a social problem in recent years is not sufficient.

A blacklist mechanism checks whether the sender's IP address and/or domain name is registered in an attacker IP address and/or domain name list, i.e., a blacklist. Blacklists provided by MxToolBox [10], Spamcop Blocking List [11], Barracuda Reputation Blocklist [12], and Spamhaus blocklist are popular. The Spamhaus blocklist, provided by The Spamhaus Project, an international non-profit organization, is the most famous and widely used IP blacklist. The Spamhaus blocklist is managed by dedicated teams in 10 countries and maintains its by tracking cyber threats such as spam, phishing, and malware, worldwide. However, blacklists have a disadvantage in that it takes time for both the removal of legitimate IP addresses from the list and the registration of malicious IP addresses to the list to be reflected in the service. For example, when an Internet service provider (ISP) sends an IP address removal request to the Spamhaus blocklist, the Spamhaus blocklist evaluates the legitimacy of the IP address, and the removal request is processed within 24 hours according to the Spamhaus blocklist policy.

B. *Sender domain authentication*

SPF, DKIM, and DMARC are popular methods of sender domain authentication. We describe these three methods in Sections II-B1, II-B2, and II-B3, respectively.

1) SPF: SPF is a method of checking the SPF record of the sender domain to make sure that the IP address of the sender's SMTP server is legitimate. Senders indicate a list of IP addresses of SMTP servers that may send e-mails from their domain as the SPF record on the Domain Name System (DNS) content server in advance. To verify an e-mail with SPF, the recipient queries the sender's Envelope-From domain's DNS content server for the SPF record to check if the SPF record contains the IP address of the sender's SMTP server. However, the verification of forwarded e-mails with this method will not be successful even for legitimate mail. Figure 2 shows an example in which SPF authentication fails for a forwarded e-mail.

As shown in this figure, when a message is forwarded, the original IP address of the SMTP server is changed to the relay

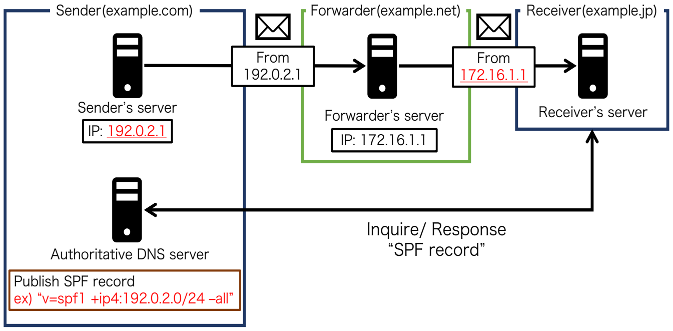


Figure 2. An example of an SPF authentication failure for a forwarded e-mail.

```
Return-Path: <sender@example.com>
(snip)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
    d=signer.example; s=20191225;
    bh=Za3JDErJrJPrpL+bXkLoOcl2gQi1jwTNEIAraa8oTDU=;
    b=yRKl7uiICDa7nBw2I0yQECGgnWWwNX+H42tMm2T4/MI/S
    6fgRL/XoOyYyNb14BtR5H7I0O8mXQKUB78cyFJj75Wy0w2RBb
    SnHTbOYM3KmEnzqu4lrFLlovRoI=
(snip)
From: <sender@example.com>
To: Receiver@example.net
```

Figure 3. An example of an e-mail header.

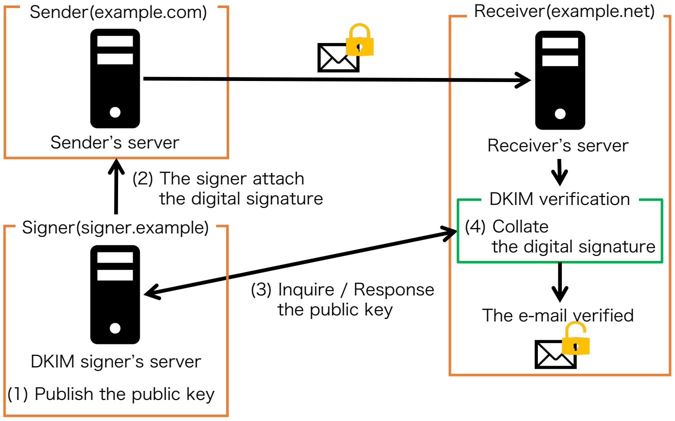


Figure 4. An example of the DKIM verification flow.

server's IP address which is not include the SPF record. As a result, there are many cases in which a valid mail fails this verification.

2) DKIM: DKIM is an authentication method that uses the digital signature generated from the body and header of an e-mail. Figure 3 shows an example of an e-mail header and Figure 4 shows an example of the DKIM verification flow.

First, to use the DKIM mechanism, the sender domain ("example.com" in Figure 4) prepares a private key and public key pair in advance and publishes the public key on their authoritative DNS server for DKIM verification ("signer.example" in Figure 4). Then, the sender domain ("example.com") generates the DKIM signature from the body and header of the e-mail using the private key and attaches it to the e-mail header as the DKIM signature, as shown by the "b=" tag in Figure 3.

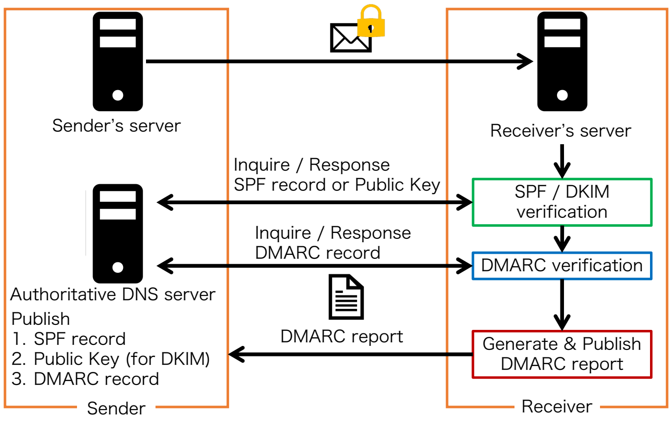


Figure 5. Flow of the DMARC verification.

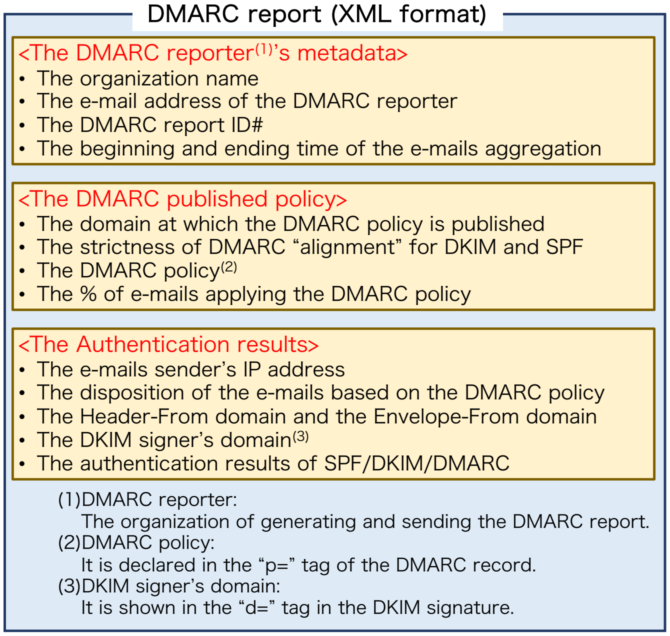
Next, the receiver ("example.net" in Figure 4) requests the public key from the sender specified domain authoritative DNS server, as is shown in the "d=" tag of the DKIM signature ("signer.example" in Figure 3 and Figure 4). Then, the receiver obtains the hash value from the digital signature using the public key and compares it to the value of the "bh=" tag of the DKIM signature. If these values are the same, the e-mail passes the DKIM verification. With this mechanism, DKIM can even correctly verify forwarded e-mail, unlike SPF.

As shown in Figure 4, the DKIM signature domain does not need to match with the name of the sender's domain. Our observations confirmed that approximately 75% of DKIM-compatible domains use a third-party signature. However, the receiver cannot distinguish whether the third-party signer is legitimate. As a result, spammers can send spoofed e-mails with a DKIM signature using their own malicious domain that will pass the verification.

Additionally, in DKIM authentication, an administrator must change the key periodically, but the key may be expired or the key information may be misdescribed. In such cases, the validation will be failed even if the e-mail delivery is legitimate.

3) DMARC: DMARC is a reporting and policy controlling framework using both the SPF and DKIM mechanisms to authenticate e-mails. Although DMARC is a relatively new technology, the adoption rate of DMARC has been increasing in recent years. One of the reasons for this is that in addition to the UK and Australian governments, the US government has also required government agencies to support DMARC [13] [14] [15]. In addition, many mail service providers (MSP), ISP, financial institutions around the world have also adopted DMARC.

Figure 5 shows the flow of the DMARC verification. To use DMARC, the sender domain administrator must publish the SPF record for SPF verification and the public key for DKIM verification on an authoritative DNS server in advance to correspond to at least one of the two authentication mechanisms. Moreover, the sender domain needs to publish the DMARC record on their DNS server. For example, when the sender domain is "example.com," a DMARC record is published as a



DMARC report (XML format)

<The DMARC reporter(1)'s metadata>
- The organization name
- The e-mail address of the DMARC reporter
- The DMARC report ID#
- The beginning and ending time of the e-mails aggregation

<The DMARC published policy>
- The domain at which the DMARC policy is published
- The strictness of DMARC "alignment" for DKIM and SPF
- The DMARC policy(2)
- The % of e-mails applying the DMARC policy

<The Authentication results>
- The e-mails sender's IP address
- The disposition of the e-mails based on the DMARC policy
- The Header-From domain and the Envelope-From domain
- The DKIM signer's domain(3)
- The authentication results of SPF/DKIM/DMARC

(1)DMARC reporter:
The organization of generating and sending the DMARC report.
(2)DMARC policy:
It is declared in the "p=" tag of the DMARC record.
(3)DKIM signer's domain:
It is shown in the "d=" tag in the DKIM signature.

Figure 6. Example of a DMARC report.

TXT record "_dmarc.example.com" under the following rules:

v=DMARC1; p=reject; rua=mailto:rua@example.net.

In the policy controlling function, DMARC provides a mechanism for the administrator of the sender domain to declare the policy for how the receiver handles an e-mail that fails sender domain authentication in the "p=" tag of the DMARC record. The value of the "p=" tag has three variations: "none" (nothing even in the case of authentication failure), "quarantine" (quarantine the authentication failure e-mail), and "reject" (reject the authentication failure e-mail).

In the reporting function, an e-mail receiver sends the DMARC report to the e-mail address of the administrator of the sender domain shown in the "rua=" tag of the DMARC record. The DMARC report provides information, such as e-mail domains, authentication results, and the effectiveness of the DMARC policy. Examples of information included in DMARC reports are shown in Figure 6. With this, the administrator of the sender domain can determine the performance of the DMARC authentication and they can take measures to prevent spoofed e-mails from abusing their domain.

According to the concept of "alignment," DMARC verification fails when the domains for SPF and DKIM verification are different from the sender's Header-From domain. The sender's Header-From domain need not be the same as the Envelope-From domain or the DKIM signature domain. However, spammers can easily imitate the Header-From domain. As a countermeasure, using alignment, the receiver can check whether the Header-From domain is correct. The sender domain can choose from two mode of alignment strictness, "strict" and "relaxed," using the DMARC record.

When the administrator of the sender domain uses the "strict" mode, DMARC verification passes only when the Header-From address and the domain for SPF or DKIM verification match completely. Conversely, when the alignment mode is "relaxed," DMARC verification will succeed if subdomains of the Header-From address and subdomains of the domain for SPF or DKIM verification match.

DMARC is one of the most effective countermeasures to spoofed e-mail. However, DMARC cannot solve the issue that SPF cannot properly verify forwarded e-mails. SPF cannot properly authenticate forwarded e-mails because the sender's IP address changes to the forwarder's IP address when the e-mails are forwarded. Moreover, because DKIM allows third-party signatures, which are commonly used worldwide, as described in Section II-B2, e-mails signed by a third-party signer will fail the DMARC verification due to alignment.

Therefore, there are cases in which legitimate forwarded e-mails will fail the DMARC authentication, e.g., when e-mails use a third-party signature or the e-mail domains are not compatible with DKIM.

C. *Content filtering*

A large number of content filtering methods have been proposed over the years. Content filtering is an effective and widely used anti-spam method. This method adapts classifiers to the content or the attached files of the e-mail. The Bayesian filter [16] [17] [18] is a well-known content filtering method using the Bayes theorem to classify the e-mail content. In addition, natural language processing [19], support vector machines [20] [21], and machine learning [22] [23] are widely used as classifiers in content filtering methods.

In actual operation, as shown in Figure 1, content filtering has a high calculation cost and is therefore used after reducing the number of e-mails to be inspected by other anti-spam methods. SpamAssassin [24] [25], for example, scores e-mails based on keywords, the public database, and a Bayesian filter to detect spam e-mails. This method uses several anti-spam methods, such as blacklist [26] [27] and sender domain authentication methods, when the e-mails are received, prior to applying the Bayesian filter.

III. Design of our method

As described in Section II, e-mail servers are generally operated using a combination of multiple anti-spam measures. In addition, sender domain authentication is processed prior to e-mail content filtering. Therefore, it is important to reduce false positives in the sender domain authentication to achieve reliable e-mail server operation. However, as described in Section II-B, sender domain authentication may mistakenly determine legitimate e-mails as spoofed e-mails in the case of forwarded e-mail, misdescription of DKIM key information, and DKIM third-party signatures, etc.

To overcome this issue, we propose a method to detect false positives generated by the existing sender domain authentications by analyzing large-scale DMARC report data using an X-means clustering analysis.

As shown in Figure 7, our method consists of the following three phases: (A) DMARC reports summarization, (B) Summarized DMARC report clustering, and (C) Legitimate senders detection.

A. *The DMARC reports summarization*

First, we describe the DMARC report aggregation ((A) in Figure 7). As described in Section II-B3, the DMARC reports

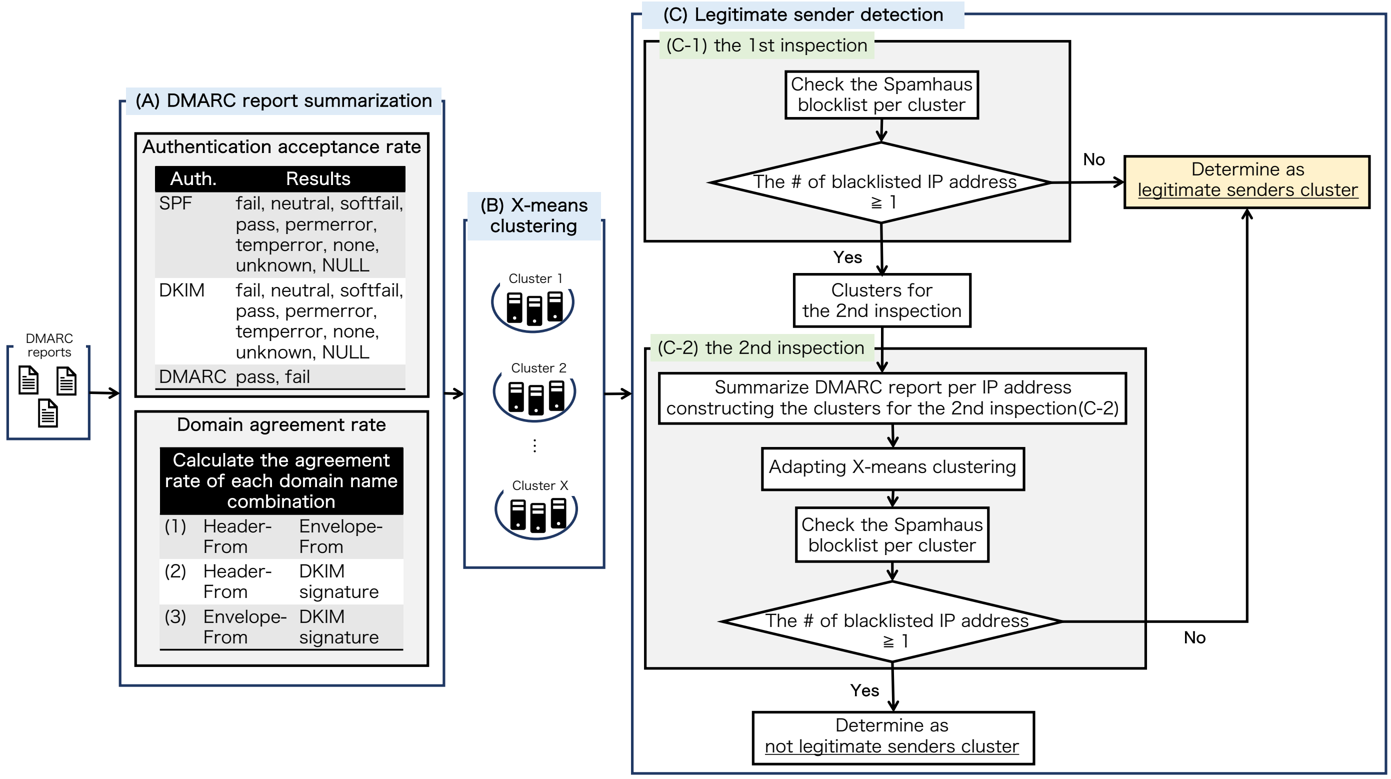

DMARC reports
(A) DMARC report summarization
Authentication acceptance rate
Auth.
Results
SPF
fail, neutral, softfail, pass, permerror, temperror, none, unknown, NULL
DKIM
fail, neutral, softfail, pass, permerror, temperror, none, unknown, NULL
DMARC
pass, fail
Domain agreement rate
Calculate the agreement rate of each domain name combination
(1) Header-From Envelope-From
(2) Header-From DKIM signature
(3) Envelope-From DKIM signature
(B) X-means clustering
Cluster 1
Cluster 2
⋮
Cluster X
(C) Legitimate sender detection
(C-1) the 1st inspection
Check the Spamhaus blocklist per cluster
The # of blacklisted IP address ≧ 1
No
Determine as legitimate senders cluster
Yes
Clusters for the 2nd inspection
(C-2) the 2nd inspection
Summarize DMARC report per IP address constructing the clusters for the 2nd inspection(C-2)
Adapting X-means clustering
Check the Spamhaus blocklist per cluster
The # of blacklisted IP address ≧ 1
No
Yes
Determine as not legitimate senders cluster

Figure 7. Overview of our method.

are provided in the XML format; therefore, it is necessary to convert the original DMARC report data into numeric data for the clustering process. Additionally, our method should summarize the DMARC reports by the sender's IP address to identify legitimate senders.

We summarize DMARC report data to allow the adaptation of a clustering analysis focusing on the results of the sender domain authentication and the e-mail domain names. As a summarization of the sender domain authentication results, we calculate the acceptance rates of SPF, DKIM, and DMARC for each IP address. Each of these three authentication methods has several authentication results, as shown in Figure 7. Our method calculates the percentage of e-mails for each authentication result per IP address.

Next, to summarize the e-mail domain names, we calculate the agreement rate for the three domain name combinations. The DMARC mechanism compares the Header-From to the Envelope-From domain ((1) in Figure 7) and the DKIM signature domain ((2) in Figure 7) for the DMARC alignment inspection. Conversely, the Envelope-From domain is not compared to the DKIM signature domain ((3) in Figure 7) in the sender domain authentication verification process. However, because we consider the combination (3), which is not for sender domain authentication, as having a relationship, it can be used to improve the accuracy of our approach.

B. Clustering the summarized DMARC reports

Second, we cluster of the aggregated DMARC reports ((B) in Figure 7). Our method adapts a clustering algorithm to the summarized DMARC report data. This clustering phase is used to classify the sender's IP addresses exhibiting similar e-mail transmission behavior trends, including those with respect to the authentication results and the consistency between the domain names related to sending e-mails with respect to the clusters. Actually, as presented in our previous study [1], we confirmed that our previous method can classify a plenty of legitimate forwarders in one cluster. Based on these results, we consider that our method can classify the sender's IP address according to the similarity associated with the e-mail sending operation in addition to the classification of the forwarding servers in our clustering phase.

We assume that the dataset is a large number of DMARC reports. Therefore, non-hierarchical clustering is better than hierarchical clustering for our method. In addition, when using a non-hierarchical clustering algorithm in our approach, we assume that it is difficult to determine the number of clusters because the scale of the DMARC reports is not constant depending on the DMARC report receiving domain. Several algorithms have been proposed that can automatically estimate the number of clusters in non-hierarchical clustering; such algorithms include affinity propagation [28], the Bayesian Gaussian mixture model [29], and X-means clustering [6]. However, to appropriately estimate the number of clusters for affinity propagation, we need to set the number of the "preference," which is the preference value for each point, depending on the dataset. Meanwhile, to estimate the appropriate number of clusters for Bayesian Gaussian mixture model, we need to adjust the parameter "reg_covar," which is a regularization added to the diagonal of the covariance, depending on the dataset. Therefore, because our method is to applied to actual DMARC reports, the affinity propagation and Bayesian Gaussian mixture model approaches are not suitable as clustering algorithms for our approach. X-means, which is also a non-hierarchical clustering approach, is a K-means extended algorithm proposed by Pelleg and Moore [6]. K-means is one of the most popular clustering methods but has a shortcoming, in which the number of clusters, K, needs to be provided by users in advance.

Conversely, X-means can determine the number of clusters, X, via iterations of K-means and splitting decisions based on the Bayesian information criterion (BIC) without complicated parameter adjustments. Accordingly, our method uses an X-means clustering analysis to classify the sender's IP address.

In the X-means clustering flow, the senders' IP addresses are divided into clusters according to their e-mail transmission behavior trends, such as the consistency between the domain names related to the e-mail sending and its authentication results.

C. Legitimate sender detection

Third, we detect the legitimate senders in our proposed approach ((C) in Figure 7). We determine legitimate senders clusters based on the Spamhaus blocklist that is the most famous IP blacklist in the world. This detection flow consists of two inspections as (C-1) and (C-2) in Figure 7.

1) The first inspection:

The first inspection ((C-1) in Figure 7) checks all of the IP addresses in all of the clusters to determine if they are listed in the Spamhaus blocklist. Then, our method classifies the clusters that do not include any IP addresses registered in the Spamhaus blocklist as legitimate senders clusters. Other clusters that have one or more IP addresses in the Spamhaus blocklist are passed to the second inspection ((C-2) in Figure 7).

2) The second inspection:

As mentioned in Section III-C1, the clusters to be checked in the second inspection ((C-2) in Figure 7) have not been determined to be clusters of legitimate senders in the first inspection (C-1) because one or more of their IP addresses are registered in the Spamhaus blocklist. In other words, these clusters consist of both blacklisted IP addresses and non-blacklisted IP addresses. As an example, let us consider the cluster that consists of one blacklisted IP address and 99 white IP addresses. As described in Section II-A, the registration, and deregistration of IP addresses on the blacklist may be delayed. Therefore, even if the e-mail sending operation of this cluster is legitimate, 99 non-blacklisted IP addresses may be affected by the one blacklisted IP address, whose deregistration from the blacklist has been delayed. Accordingly, our method performs a second inspection (C-2) to further improve the false positive detection performance. However, because these clusters actually contain at least one or more blacklisted IP addresses, our method cannot use these clusters to detect legitimate servers. Therefore, as the second inspection (C-2), our method performs the following clustering to detect additional legitimate servers.

As the first step in the second inspection (C-2), our method adapts the DMARC report summarization phase and X-means clustering phase to the IP addresses constructing clusters in the same way as in (A) and (B) in Figure 7. Then, as in the first inspection (C-1), our method checks the Spamhaus blocklist to determine whether one or more IP addresses are listed for each cluster. Further, as with the first inspection (C-1), if no IP addresses are listed in the Spamhaus blocklist, our method classifies the clusters as legitimate senders clusters. Otherwise, our method determines these clusters to be non-legitimate senders clusters.

TABLE I. LIST OF ABBREVIATIONS USED IN THIS PAPER.

Abbreviations	Details
Day	Day the DMARC report received
All_IP	The total number of sender server IP addresses in the DMARC reports
All_mail	The total number of e-mails constructing the DMARC reports
All_rep	The total number of DMARC reports
Tgt_IP	The IP addresses adapting to our method *These IP addresses send 90% of all e-mails constructing the DMARC reports.
1st_Tgt_IP	The IP addresses for X-means in the first inspection ((C-1) in Figure 7) *These IP addresses are same as those in Tgt_IP.
2nd_Tgt_IP	The IP addresses for X-means in the second inspection (C-2 in Figure 7) *These IP addresses are not detected as legitimate senders in the first inspection ((C-1) in Figure 7).
1st_C	The number of clusters as the clustering result of the first inspection ((C-1) in Figure 7)
2nd_C	The number of clusters as the clustering result of the second inspection ((C-2) in Figure 7)
1st_Leg_C	The legitimate sender clusters detected in the first inspection ((C-1) in Figure 7)
2nd_Leg_C	The legitimate sender clusters detected in the second inspection ((C-2) in Figure 7)
1st_Leg_IP	The legitimate IP addresses in *Leg_C* detected in the first inspection ((C-1) in Figure 7)
2nd_Leg_IP	The legitimate IP addresses in *Leg_C* detected in the second inspection ((C-2) in Figure 7)
Leg_IP	All legitimate IP addresses detected by our method *(the # of *Leg_IP*) = (the # of *1st_Leg_IP*) + (the # of *2nd_Leg_IP*)

IV. DATASET

In this section, we describe the dataset we used to test our method. We use the actual DMARC reports received from November 1 to November 30, 2019, at one of the most famous ISP domains in Japan.

The abbreviations that we use in the following discussion and in the results are shown in Table I. Table II shows the number of sender IP addresses ("*All_IP*"), DMARC reports ("*All_rep*"), and e-mails ("*All_mail*") in the DMARC report dataset used in this experiment. As shown in the bottom-most row of the "*All_rep*" column in Table II, we observed 74,199 DMARC reports on average (45,884–100,536). These DMARC reports are constructed by 501,927 e-mails on average (385,115–637,727) that sent from 11,418 sender IP addresses on average (7,390–19,330), as shown in the bottom-most row of the "*All_mail*" and the "*All_IP*" column in Table II, respectively. The "*# of Tgt_IP*" and the "*Tgt_IP/All_IP (%)*" columns in Table II show the number and the ratio to the "*All_IP*" of the sender IP addresses that we apply to our method.

TABLE II. UTILIZED DATASET.

Day	*All_IP*	*All_mail*	*All_rep*	*# of Tgt_IP*	*Tgt_IP / All_IP* (%)
Day 1	12,614	438,216	59,794	1,804	14.3
Day 2	10,314	420,850	51,527	1,346	13.1
Day 3	9,445	494,184	53,033	1,164	12.3
Day 4	7,390	436,984	45,884	1,074	14.5
Day 5	7,641	447,544	46,655	1,100	14.4
Day 6	10,839	592,334	59,038	1,242	11.5
Day 7	11,617	495,553	65,319	1,703	14.7
Day 8	10,996	495,411	67,184	1,812	16.5
Day 9	11,624	491,806	75,665	2,080	17.9
Day 10	9,757	486,201	71,167	2,030	20.8
Day 11	8,013	402,857	65,537	1,972	24.6
Day 12	11,297	510,453	79,405	2,228	19.7
Day 13	12,789	561,485	86,690	2,469	19.3
Day 14	12,584	588,425	92,324	2,537	20.2
Day 15	12,014	626,296	85,930	2,399	20.0
Day 16	11,835	554,598	83,702	2,468	20.9
Day 17	9,796	428,524	78,384	2,428	24.8
Day 18	8,381	385,115	73,551	2,389	28.5
Day 19	11,323	520,894	79,461	2,319	20.5
Day 20	12,179	456,908	76,983	2,478	20.3
Day 21	19,330	637,727	100,536	3,149	16.3
Day 22	13,891	543,432	90,515	2,897	20.9
Day 23	12,027	488,000	81,262	2,714	22.6
Day 24	11,372	506,560	74,459	2,318	20.4
Day 25	8,773	386,306	65,246	2,315	26.4
Day 26	11,523	507,158	77,065	2,517	21.8
Day 27	12,520	493,725	78,698	2,667	21.3
Day 28	15,950	567,088	98,018	3,007	18.9
Day 29	12,467	518,344	81,857	2,717	21.8
Day 30	12,250	574,834	81,073	2,562	20.9
Minimum	7,390	385,115	45,884	1,074	11.5
Maximum	19,330	637,727	100,536	3,149	28.5
Average	11,418	501,927	74,199	2,197	19.3

As shown in the bottom-most row of Table II, the number of *Tgt_IP* accounts for 19.3% on average (11.5–28.5%) of *All_IP*. According to our observations in Table II, *Tgt_IP* sends more than 90% of the e-mails of *All_mail*. By contrast, the remaining IP addresses, which are not *Tgt_IP*, send less than 10% of the e-mails of *All_mail*. Because these remaining IP addresses, which send only a few e-mails, will constitute noise for the X-means clustering algorithm, we utilize only the DMARC reports, for which the senders' IP addresses are included in *Tgt_IP*.

V. RESULTS

In this section, we explain the results obtained by applying our method to the dataset described in Section IV. Table III shows the results of the X-means clustering and legitimate IP address detection.

First, the average number of IP addresses for the first inspection (*1st_Tgt_IP*) is 2,197 per day (1,074–3,149). In the first inspection ((C-1) in Figure 7), our method divided *1st_Tgt_IP* into 20 clusters on all days, as shown in the "*1st_C*" column in Table III. As the result of the first inspection (C-1), the number of legitimate sender clusters (*1st_Leg_C*) was 15 per day on average (12–17), as shown in the "# of *1st_Leg_C*." Moreover, 324 IP addresses per day on average (164–493) were contained within *1st_Leg_C*, as shown in the "# of *1st_Leg_IP*" column in Table III. Then, as described in Section III, the second inspection ((C-2) in Figure 7) applied DMARC report summarization, X-means clustering, and legitimate sender detection to the clusters for the second inspection (*2nd_Tgt_IP*, which was generated by the first inspection (C-1). As shown in the "# of *2nd_Tgt_IP*" column in Table III, the number of IP addresses subject to the second inspection (*2nd_Tgt_IP*) was 1,873 per day on average (910–2,753). In the second inspection (C-2), our method classified *2nd_Tgt_IP* into 20 clusters on all days, as shown in the "*2nd_C*" column in Table III. The second inspection determined 5 clusters on average (1–11) to be legitimate sender clusters (the "# of *2nd_Leg_C*" column in Table III). In addition, *2nd_Leg_C* consisted of 23 IP addresses on average (1–133), as shown in the "# of *2nd_Leg_IP*" column in Table III.

As a result of applying our method to the dataset, our method detected 347 legitimate senders' IP addresses per day on average (178–732), as shown in the "# of *Leg_IP*" column in Table III.

VI. EVALUATION

In this section, we evaluate the results of applying our method, as described in Section V, to the dataset. As described in Section III, none of the legitimate IP addresses detected in our method are included in the Spamhaus blocklist. This means that none of the legitimate IP addresses detected using our approach were the known spammer IP addresses.

Forwarded e-mails are prone to false positives with sender domain authentications, as described in Section II-B. To determine if our method successfully detected forwarded e-mails as legitimate senders, we confirmed the classification results of five IP addresses that were known forwarding servers in the domain of the ISP that received the DMARC reports used as the dataset. We confirmed that these five IP addresses were successfully classified in the same cluster, which was detected as a legitimate sender cluster by our method.

Then, we evaluated the results focusing on the following two points: the detected legitimate e-mails (A) and the detected false positive deliveries with respect to the sender domain authentications (B).

A. *The detected legitimate e-mails*

First, we checked the number of e-mails sent from the IP addresses (*Leg_IP*, which consisted of 347 servers on average, as shown in Table III) of the servers that our method detected as legitimate senders.

Figure 8 shows the number of legitimate e-mails sent from the legitimate IP addresses detected by our approach. As shown in this figure, combining the first and second inspections, our method detected 214,153 legitimate e-mails per day on average (110,484–340,473). From this result, we confirmed that our method can detect a large number of legitimate e-mails in the sender authentication.

As mentioned in Section II-A, blacklist techniques have an issue in that both the registration and deregistration of IP addresses is delayed. This delay can cause many non-blacklisted IP addresses to be incorrectly classified into the same cluster as a few blacklisted IP addresses, as we described in Section III-C2. To counter this problem, our method performs a second inspection after the first inspection, as described in Section III-C2. As we can see from Figure 8, which shows the number of legitimate e-mails detected by our method, the second inspection in our method was able to detect 20,141 additional legitimate e-mails per day on average (146–116,888). In particular, for example on *Day* 22

TABLE III. THE RESULTS OF APPLYING OUR METHOD TO THE DATASET.

Day	# of *1st_Tgt_IP*	*1st_C*	# of *1st_Leg_C*	# of *1st_Leg_IP*	# of *2nd_Tgt_IP*	*2nd_C*	# of *2nd_Leg_C*	# of *2nd_Leg_IP*	# of *Leg_IP*
Day 1	1,804	20	17	694	1,110	20	6	38	732
Day 2	1,346	20	16	431	915	20	2	4	435
Day 3	1,164	20	15	199	965	20	3	7	206
Day 4	1,074	20	16	164	910	20	10	14	178
Day 5	1,100	20	17	175	925	20	3	7	182
Day 6	1,242	20	16	277	965	20	5	8	285
Day 7	1,703	20	17	370	1,333	20	4	13	383
Day 8	1,812	20	17	365	1,447	20	4	15	380
Day 9	2,080	20	12	208	1,872	20	8	99	307
Day 10	2,030	20	14	260	1,770	20	1	1	261
Day 11	1,972	20	15	192	1,780	20	7	28	220
Day 12	2,228	20	17	369	1,859	20	1	2	371
Day 13	2,469	20	17	378	2,091	20	5	17	395
Day 14	2,537	20	14	327	2,210	20	2	2	329
Day 15	2,399	20	16	326	2,073	20	3	26	352
Day 16	2,468	20	14	316	2,152	20	6	24	340
Day 17	2,428	20	13	215	2,213	20	7	35	250
Day 18	2,389	20	16	244	2,145	20	2	9	253
Day 19	2,319	20	17	336	1,983	20	11	45	381
Day 20	2,478	20	17	493	1,985	20	1	7	500
Day 21	3,149	20	15	396	2,753	20	4	10	406
Day 22	2,897	20	14	166	2,731	20	9	133	299
Day 23	2,714	20	16	371	2,343	20	3	9	380
Day 24	2,318	20	14	318	2,000	20	6	21	339
Day 25	2,315	20	14	279	2,036	20	1	4	283
Day 26	2,517	20	16	327	2,190	20	5	11	338
Day 27	2,667	20	15	348	2,319	20	5	29	377
Day 28	3,007	20	14	396	2,611	20	5	41	437
Day 29	2,717	20	15	392	2,325	20	4	12	404
Day 30	2,562	20	15	379	2,183	20	4	18	397
Minimum	1,074	20	12	164	910	20	1	1	178
Maximum	3,149	20	17	694	2,753	20	11	133	732
Average	2,197	20	15	324	1,873	20	5	23	347

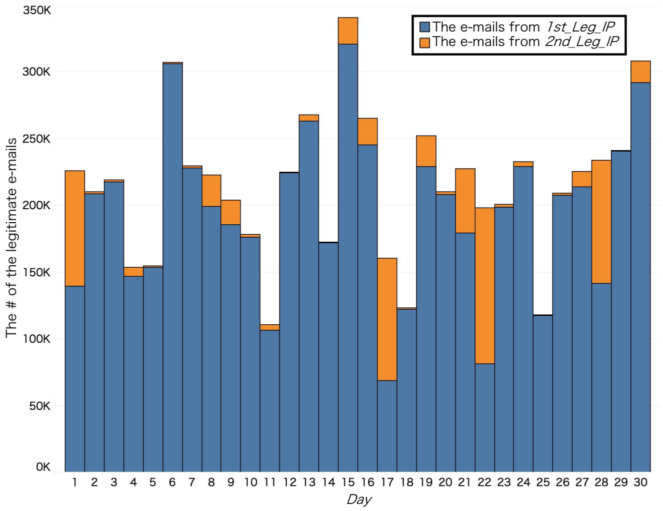


Figure 8. The number of e-mails sent from the legitimate IP addresses detected by our method.

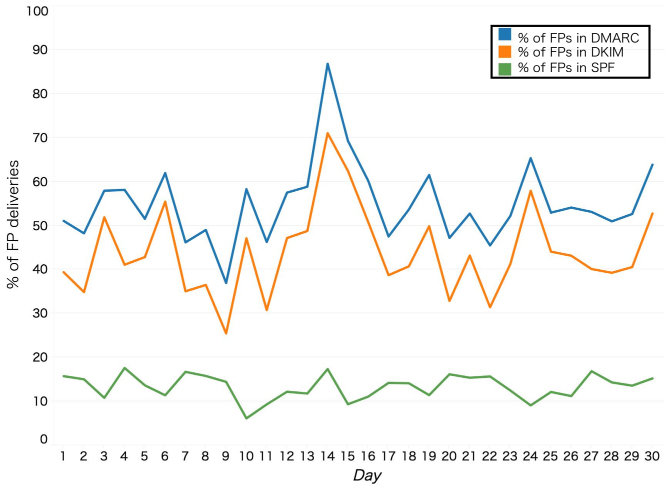
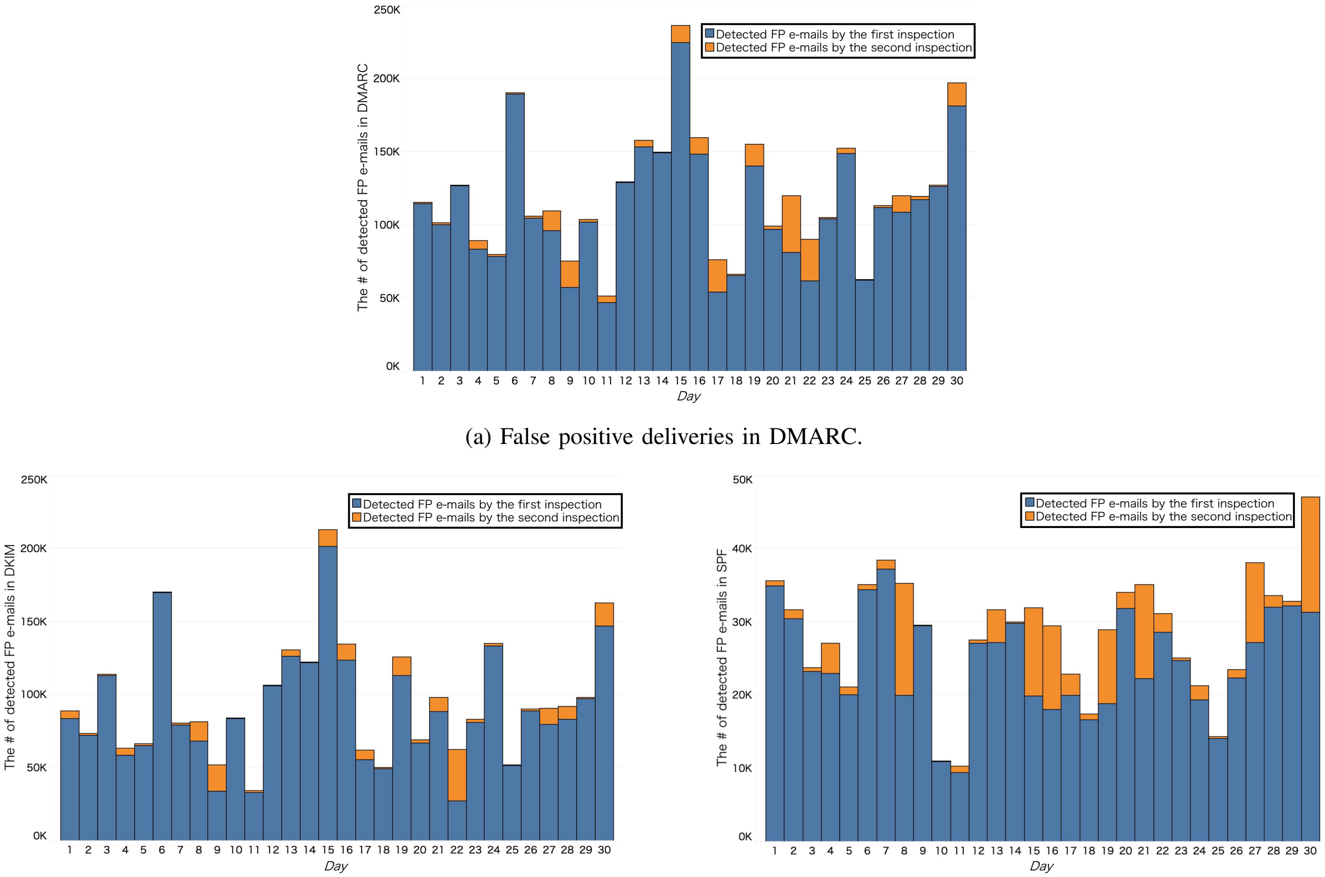


Figure 9. The percentage of false positive deliveries in the sender domain authentication for e-mail deliveries from the legitimate senders detected by our method.

in Figure 8, the first inspection found 81,375 legitimate e-mails, while the second inspection found an additional 116,888 legitimate e-mails. In other words, approximately 59.0% of the legitimate e-mails detected on that day were detected by the second inspection. These results show that the second inspection was able to detect many legitimate senders that were incorrectly classified as non-legitimate sender clusters during the first inspection. Therefore, we confirmed that our method can improve the detection performance by performing a second inspection in addition to the first inspection.

The detected legitimate e-mails contain e-mails that both failed and passed sender domain authentications. Figure 9 shows the ratio of e-mails that failed the SPF, DKIM, and DMARC authentications to the total legitimate e-mails shown in Figure 8 for each day. As shown by the blue line in Figure 9, the ratio of DMARC failed e-mails was 55.1% on average (37.0–86.8%). The orange line in Figure 9 shows the ratio



(a) False positive deliveries in DMARC.

(b) False positive deliveries in DKIM.

(c) False positive deliveries in SPF.

Figure 10. The number of false positive deliveries detected by our approach.

of DKIM failed e-mails, which was 43.9% on average (25.5–71.0%). Meanwhile, the green line in Figure 9 indicates that the ratio of SPF failed e-mails to legitimate e-mails was 13.4% on average (6.2–17.6%).

According to these results, a large number of sender domain authentication failure e-mails are contained in the detected legitimate e-mails. We consider these sender domain authentication failure e-mails in detail in Section VI-B.

B. The detected false positive deliveries in the sender domain authentications

As mentioned in Section VI-A, because the legitimate e-mails detected by our method were sent from legitimate IP addresses, the legitimate e-mails that failed SPF, DKIM, and DMARC are false positives in the sender domain authentications.

In this section, we investigate the number of sender domain authentication failure deliveries, that is, the false positive deliveries, for each method of sender domain authentication.

Figure 10 shows the number of detected false positives for the SPF, DKIM, and DMARC authentications. As shown in Figure 10(a), 119,405 legitimate e-mails on average (51,132–235,908) were detected as false positives in the DMARC authentication by our method, including both the first and the second inspections. In addition, as shown in Figure 10(b), our method detects 96,129 legitimate e-mails on average (34,051–212,661) as false positives in the DKIM authentication. Meanwhile, Figure 10(c) indicates that our method detected 28,466 legitimate e-mails on average (10,322–47,010) as false positives in the SPF authentication.

From these results, we confirmed that our method is able to detect various types of deliveries that are false positives in sender domain authentication without using e-mail contents. By utilizing the proposed method, e-mail system administrators can significantly reduce the false positives that occur with conventional sender domain authentication.

VII. Conclusions

In general anti-spam operation, e-mails are inspected by sender domain authentications prior to content filtering. Therefore, it is critical to reduce the false positives in the sender domain authentication, as opposed to the false negatives, to enable reliable e-mail server operation.

In this paper, we proposed a method to detect false positives generated by existing sender domain authentications by analyzing massive amounts of DMARC report data using an X-means clustering analysis.

Our approach consisted of three phases: DMARC report summarization, X-means clustering, and legitimate sender detection.

In the DMARC report summarization, our approach summarized the DMARC reports for each e-mail sender's IP address focusing on the results of the sender domain authentications and combinations of the Header-From domain, the Envelope-From domain, and the DKIM signature domain.

Then, our approach adapted X-means clustering to the summarized DMARC reports to classify the e-mail sender's IP address based on transmission behavior, such as the consistency between the domain names related to e-mail sending and its authentication.

Next, our approach detected the legitimate sender clusters by processing two inspections. The first inspection checked whether the IP addresses in the clusters were included in the Spamhaus blocklist. If no IP addresses in the cluster were included in the Spamhaus blocklist, the first inspection determined the cluster to be a legitimate sender cluster. The other clusters consisted of both blacklisted IP addresses and non-blacklisted IP addresses. However, non-blacklisted IP addresses may be incorrectly classified into the same cluster as a few blacklisted IP addresses because both the registration and deregistration of IP addresses in the blacklist are not processed immediately. Therefore, to improve the performance of the legitimate sender detection, a second inspection checked the clusters that were not determined to be legitimate sender clusters in the first inspection.

In the second inspection, as in the first inspection, our method aggregated the DMARC reports for the IP addresses that were subject to the second inspection, performed X-means clustering, and determined the validity of the clusters using the Spamhaus blocklist.

We applied our method to actual DMARC report data and detected 214,153 e-mails on average (110,484–340,473) sent from 347 legitimate senders' IP addresses on average (178–732) as legitimate e-mails per day.

In addition, to evaluate the effect of reducing the false positives that occur in the sender domain authentication when using our method, we investigated the percentage of e-mails sent from the legitimate sender addresses that failed sender domain authentications using the DMARC reports. As a result, we confirmed that, on average, 13.4% (6.2–17.6%), 43.9% (25.5–71.0%), and 55.1% (37.0–86.8%) false positives occurred when using SPF, DKIM, and DMARC, respectively. This result shows that our method detects false positive e-mails in conventional anti-spam systems by detecting e-mails, such as forwarded e-mails, which cannot be correctly classified by existing sender authentication technologies.

Our method does not use e-mail contents, only DMARC report data, and can effectively detect deliveries that would be false positives with conventional sender domain authentications. In addition, since this method can be operated independently of the sending and receiving resources of the e-mail system, it can be installed without increasing the load on the entire e-mail system.

The evaluation for accuracy of our method when our method is operated continuously in actual large scale e-mail system is our future work.

Acknowledgments

This paper was written based on the research while Kanako Konno and Naoya Kitagawa, who the authors of this paper, were with Tokyo University of Agriculture and Technology, Japan. The authors would like to thank TwoFive, Inc. for supporting this study. The authors would also like to thank Mr. Shuji Sakuraba (Internet Initiative Japan Inc.) for constructive discussion.

References

[1] K. Konno, N. Kitagawa, S. Sakuraba, and N. Yamai, "Legitimate E-mail Forwarding Server Detection Method by X-means Clustering Utilizing DMARC Reports," in the Eleventh International Conference on Evolving Internet (INTERNET 2019), 2019, pp. 24–29.

[2] FBI (Federal Bureau of Investigation), "Business email compromise the $26 billion scam," 2019, [Online]. Available: https://www.ic3.gov/media/2019/190910.aspx [Accessed: 1st Jun. 2020].

[3] M. Wong and W. Schlitt, "Sender Policy Framework (SPF) for authorizing use of domains in e-mail," 2006.

[4] D. Crocker, T. Hansen, and M. Kucherawy, "DomainKeys Identified Mail (DKIM) signatures," sep 2011.

[5] M. Kucherawy and E. Zwicky, "Domain-based message authentication, reporting, and conformance (DMARC)," 2015.

[6] D. Pelleg and A. Moore, "X-means: Extending K-means with Efficient Estimation of the Number of Clusters," in Proceedings of the 17th International Conference on Machine Learning, 2000, pp. 727–734.

[7] E. Harris, "The Next Step in the Spam Control War: Greylisting," 2019, [Online]. Available: http://projects.puremagic.com/greylisting/whitepaper.html [Accessed: 1st Jun. 2019].

[8] T. Hunter, P. Terry, and A. Judge, "Distributed tarpitting: Impeding spam across multiple servers," in Proceedings of the 17th Large Installation Systems Administration, vol. 3, 2003, pp. 223–236.

[9] N. Kitagawa, H. Takakura, and T. Suzuki, "An anti-spam method via real-time retransmission detection," in Proceedings of 18th IEEE International Conference on Networks (ICON), 2012, pp. 382–388.

[10] MxToolbox, Inc., "MxToolbox," [Online]. Available: https://mxtoolbox.com/SuperTool.aspx [Accessed: 1st Jun. 2020].

[11] Cisco Systems, Inc, "SpamCop Blocking List," [Online]. Available: https://www.spamcop.net/bl.shtml [Accessed: 1st Jun. 2020].

[12] Barracuda Networks., "Barracuda Reputation Block List (BRBL)," [Online]. Available: http://www.barracudacentral.org/rbl [Accessed: 1st Jun. 2020].

[13] S. M. Jones, "DMARC Required For UK Government Services By October 1st," 2016, [Online]. Available: https://dmarc.org/2016/06/dmarc-required-for-uk-government-services-by-october-1st/ [Accessed: 1st Jun 2020].

[14] S. M. Jones, "Australian Government Agency Recommends DMARC, DKIM, and SPF," 2016, [Online]. Available: https://dmarc.org/2016/08/australian-government-agency-recommends-dmarc-dkim-and-spf/ [Accessed: 1st Jun. 2020].

[15] U.S. Department of Homeland Security, "Binding Operational Directive 18-01 – Enhance Email and Web Security," 2017, [Online]. Available: https://cyber.dhs.gov/bod/18-01/ [Accessed: 1st Jun. 2020].

[16] P. Graham, "A plan for spam," 2002, [Online]. Available: http://www.paulgraham.com/spam.html [Accessed: 22th Feb. 2020].

[17] P. Graham, "Better bayesian filtering," 2003, [Online]. Available: http://www.paulgraham.com/better.html [Accessed: 22th Feb. 2020].

[18] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos, "An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages," in Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000, pp. 160–167.

[19] S. Aggarwal, V. Kumar, and S. D. Sudarsan, "Identification and detection of phishing emails using natural language processing techniques," in Proceedings of the 7th International Conference on Security of Information and Networks, 2014, pp. 217–222.

[20] H. Ducker, D. Wy, and V. N. Vapnik, "Support vector machines for spam categorization," IEEE Transactions on Neural networks, vol. 10, no. 5, pp. 1048–1054, 1999.

[21] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, “A support vector machine based naive Bayes algorithm for spam filtering,” in 2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC). IEEE, 2016, pp. 1–8.

[22] T. S. Guzella and W. M. Caminhas, “A review of machine learning approaches to spam filtering,” Expert Systems with Applications, vol. 36, no. 7, pp. 10 206–10 222, 2009.

[23] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. A. Najada, “Survey of review spam detection using machine learning techniques,” Journal of Big Data, vol. 2, no. 1, p. 23, 2015.

[24] “The Apache SpamAssassin Project,” [Online]. Available: http://spamassassin.apache.org/ [Accessed: 1st Jun. 2020].

[25] J. Mason, “Filtering spam with spamassassin,” 2002.

[26] S. Sinha, M. Bailey, and F. Jahanian, “Shades of Grey: On the effectiveness of reputation-based “blacklists”,” in 2008 3rd International Conference on Malicious and Unwanted Software (MALWARE), 2008, pp. 57–64.

[27] C. J. Dietrich and C. Rossow, “Empirical research of ip blacklists,” in ISSE 2008 Securing Electronic Business Processes. Springer, 2009, pp. 163–171.

[28] B. J. Frey and D. Dueck, “Clustering by passing messages between data points,” Science, vol. 315, no. 5814, pp. 972–976, 2007.

[29] H. Attias, “A variational baysian framework for graphical models,” in Advances in Neural Information Processing Systems 12, S. A. Solla, T. K. Leen, and K. Müller, Eds. MIT Press, 2000, pp. 209–215.