Proceedings of the APAN – Research Workshop 2016
ISBN 978-4-9905448-6-7

Design and Implementation of a DMARC Verification Result Notification System

Naoya Kitagawa, Toshiki Tanaka, Masami Fukuyama and Nariyoshi Yamai

***Abstract*—Damages caused by spoofed e-mails as sent from a bank, a public organization and so on become serious social problems. In such e-mails attackers forge the sender address to defraud receivers of their personal and/or secret information. As a countermeasure against spoofed e-mails, sender domain authentication methods such as SPF and DKIM are frequently utilized. However, since most spoofed e-mails do not include DKIM signature in their e-mail header, those e-mails cannot be authenticated by the conventional system. Additionally DKIM has a problem that cannot determine whether the attached signature is legitimate. In this paper, we propose a method to detect spoofed e-mails and alert the user without DKIM signature by utilizing DMARC and implement a system that sends DMARC verification results to receivers. By utilizing this system, the users can obtain alerts for spoofed e-mails that the existing systems cannot warn.**

***Index Terms*—Anti spam, DKIM, DMARC, Sender Domain Authentication, SPF, Spoofed E-mail**

I. Introduction

E-mail communication is one of the most widely used service on the Internet. However, various malicious usages of e-mail have been becoming a serious social problem over the years. For instance, MITM (Man In The Middle) attack and DDoS (Distributed Denial of Service) attack are typical abuse examples of e-mail communication. In addition, phishing mails, that aim to defraud receivers of their personal and/or secret information under the guise of a bank or a public organization and so on, are frequently in circulation. Such e-mails are called spoofed e-mails since the most senders spoof their addresses or display names. Moreover, the damages have been growing by fraud caused by spoofed e-mails, therefore many police agencies around the world such as the FBI have been alerting [1].

Sender domain authentication methods have been proposed as countermeasure mechanisms against spoofed e-mails. As typical sender domain authentication method, SPF (Sender Policy Framework) [2] and DKIM (DomainKeys Identified Mail) [3] are widely utilized. SPF examines the validity of the sending mail server using the IP address. DKIM examines whether the message has not been tampered and whether the message has sent from proper sender using the digital signature. However, since most spoofed e-mails are considered to be sent without DKIM signature in the mail header, they cannot be verified by DKIM.

In this paper, we propose a method to distinguish spoofed e-mails without DKIM signature by using DMARC (Domain-based Message Authentication, Reporting, and Conformance) [4]. Although DMARC is utilized for the administrator of sender's domain to obtain the aggregate report or authentication failure report in general, our system notifies the receivers of spoofed e-mail by utilizing DMARC. To realize this method, we implemented a system that performs sender domain authentication using DMARC, and notifies the receiver of the authentication result according to the contents of DMARC policy to each receivers.

The rest of the paper is organized as follows. In Section II, we present existing methods. In Section III, we describe the design of our spoofed e-mail alert system. Then, Section IV shows an implementation method of the system. Section V shows notification examples of DMARC verification results and an alert example of an actual received spoofed e-mail. Finally, we present concluding remarks and suggestions for future study.

Submitted Date: 27th May 2016.

Naoya Kitagawa is with Division of Advanced Information Technology & Computer Science, Department of Institute of Engineering, Tokyo University of Agriculture and Technology, Koganei, Tokyo 184-0026 Japan (e-mail: nakit@cc.tuat.ac.jp).

Toshiki Tanaka is with Department of Computer and Information Sciences, Faculty of Engineering, Tokyo University of Agriculture and Technology, Koganei, Tokyo 184-0026 Japan (e-mail: t-tanaka@net.cs.tuat.ac.jp).

Masami Fukuyama is with Department of Computer and Information Sciences, Graduate School of Engineering, Tokyo University of Agriculture and Technology, Koganei, Tokyo 184-0026 Japan (e-mail: mfuk@net.cs.tuat.ac.jp).

Nariyoshi Yamai is with Division of Advanced Information Technology & Computer Science, Department of Institute of Engineering, Tokyo University of Agriculture and Technology, Koganei, Tokyo 184-0026 Japan (email: nyamai@cc.tuat.ac.jp).

```
Return-Path: <sender@example.com>
(snip)
DKIM-Signature:v=1; a=rsa-sha256;
      c=relaxed/relaxed;
      d=example.net;
      s=20120113;
      h=mime-version:date:message-id:subject:from:to:
      content-type;
      bh=YzODIQzFL5CIwg3H6lYD6ZafgsQR/7HxA6gRkSc7Vvg=;
      b=Jd6cf0fJGsMyekr7dUL6jjxVywqRXhkKeBcdFYdk/KzuHKZisyg/3
      iJMNlQq7wtDT6wU9uijAoEnPQirUwCHLFCJHqkliiDBva56Ec5nuGX
      AxsjLCU3XwwMQ1ABcGSepSl+e5kozZFBG7ItOZ5eXBXEyAAvChoLgu
      jjnUHJtS6uYOuSC6pVlHpyg1uzm+bVk97/w0dxc64Z8xaWMneN6KBL
      od28r7KORNgU8K6GKkwjfcYi1lkm1KBuW3X9YR8nVmhXjsRIyEhz25
      6a3WLYqKbC7cPHaK8lxFVHzE1AoZwhsgMRCswRCR9026OkWSvpuVvk
      +qN5CsarxWxmA==
(snip)
From: <sender@example.com>
To: receiver@example.org
```

Fig. 1. A sample of E-mail Header

II. EXISTING METHODS

A. Sender Domain Authentication

Currently, SPF and DKIM have been widely utilized as sender domain authentication methods.

SPF is an authentication method using the IP address of the sender's SMTP (Simple Mail Transfer Protocol) server and the domain of Envelope-From address. In order to use the verification method, a sender domain publishes an SPF record at its own DNS (Domain Name System) server in advance. The SPF record indicates the servers that may send messages with the sender address of the domain. Then, a receiver obtains the sender's SPF record and investigates whether the IP address of the sender's SMTP server is included in the SPF record. However, SPF has a problem that is not able to authenticate forwarded messages properly. This is because the IP address of the SMTP server becomes the IP address of the relay server rather than that of the original server is used for authentication, which does not match the SPF record.

Secondly, DKIM is a method using digital signature. In order to use this method, a sender domain prepares a pair of a private key and a public key in advance. Then, the sender domain publishes the public key at their DNS server. At the time of mail sending, the sender domain creates a digital signature from the mail header and the body using the secret key, and adds it to the mail header as the DKIM signature. In Fig.1, the value of "b=" tag shows the DKIM signature. Then, a receiver queries the public key to the authoritative DNS server of sender's domain that obtained from the "d=" tag of DKIM signature header. Subsequently the receiver compares the hash value obtained from the digital signature by using the public key with the hash value, that is the value of "bh=" tag. As a result, DKIM verification will be success when these values are the same. With such a mechanism, unlike SPF, DKIM can verify forwarded messages properly.

However, DKIM permits even a "d=" tag domain (example.net in Fig.1) different from the domain of Envelope-From address (example.com in Fig.1). Thus, if a spammer sends spoofed e-mails from the address of his/her own domain with the DKIM signature, the DKIM verification will be success.

B. DKIM Verification System Using POP Proxy

Our research group has proposed a system to perform a sender domain authentication by DKIM using a POP proxy [5]. Although DKIM verification is usually performed by mail service provider's server, this system verifies messages by using a POP proxy installed by each organization. In addition the system reports the verification results to each user. Even if the receiving mail server that is operated by universities, companies, ISPs, and so on does not support DKIM verification, the verification gets available independently by introducing this system at each organization. In this system, when the proxy receives a retrieval request from a mail client, the proxy gets messages from the mail server and performs DKIM verification. Then, the proxy puts the verification result into the mail header. Based on the verification result, proxy or MUA (Mail User Agent), such as Outlook, notifies the result to each user. Of course, since this system notifies based on DKIM verification result, the system cannot perform the verification for the messages without DKIM signature.

C. DMARC

DMARC is a framework of reporting and declaration of policy control using two sender domain authentication mechanisms, SPF and DKIM, and that has been spreading recently. The reporting function notifies the authentication failure reports and the aggregate reports to the administrator of the sender's domain. The administrator is able to know whether the authentication has been performed as intended by this report. On the other hand, in the policy declaration function, a sender can specify the e-mails handling method in case of sender domain authentication failure.

In addition, DMARC has the concept of "alignment". This concept means that DMARC verification gets failed even if the domain for verification (SPF and DKIM) is different from the sender's Header-From domain. SPF and DKIM verification need not be the same the Header-From domain and Envelope-From domain for SPF or the domain for signature for DKIM. Moreover, attackers can spoof the Header-From address easily. By taking advantage of alignment, a receiver can confirm the validity of the Header-From domain. A sender domain can specify the strictness of relationships between these domains using DMARC record. If a sender specifies "strict" as the alignment, DMARC verification will fail unless the domain of the Header-From address and the domain for SPF or DKIM verification match completely. On the other hand, if a sender specifies "relaxed" as the alignment, DMARC verification will succeed if the subdomains of the domain are identical.

In order to use this mechanism, a sender domain needs to support SPF and/or DKIM. Additionally, the sender domain must publish the DMARC record at its DNS contents server. DMARC record shows the recipient e-mail address for

TABLE I
VALUES OF "P=" TAG AND CORRESPONDING HANDLING METHODS

"p=" tag	How to Handle failed-messages
None	Inaction even if the authentication failed.
Quarantine	Quarantine the authentication failure mails.
Reject	Do not receive the authentication failure mails.

verification result reports and indicates the e-mails handling method in case of sender domain authentication failure. A receiver domain performs sender domain authentication of both SPF and DKIM, and applies the policy when the both of verifications are failed.

As mentioned above, a sender domain specifies the handling method for the verification-failed messages at the "p=" tag of DMARC record as the DNS contents server. TABLE I shows the values of "p=" tag and processing details corresponding to the each policy.

For example, let us consider the case where a sender domain (example.com) is supporting DMARC. Then, we assume that the sender domain is publishing the DMARC record as a TXT record of "_dmarc.example.com" in the following manner.

v=DMARC1\; p=none\; rua=mailto:reports@example.com

In this example, since the value of "p=" tag is "none", the administrator of example.com requests not to perform the isolation or reception rejection of the e-mails even if the DMARC verification is failed. Additionally, the administrator requests to send the reports to "reports@example.com" as shown in the "rua=" tag.

Fig.2 shows the flow of DMARC verification.

1) A sender domain supports own domain to the SPF and/or DKIM.
2) The sender domain also publishes the DMARC record as a TXT record of its DNS contents server.
3) SPF and DKIM verifier on receiver mail server sends a query to the DNS contents server and gets the SPF record and the public key for DKIM. Then it performs the SPF and the DKIM verification.
4) SPF and DKIM verifier sends the verification results to the DMARC verifier.
5) DMARC verifier sends a query for DMARC record to the DNS contents server of the Header-From domain.
6) If DMARC verifier obtains the DMARC record, it applies the DMARC policy based on the verification results of SPF and DKIM, and whether the sender domain matches the "alignment".
7) DMARC verifier creates an aggregate report containing the verification results and the applied policy, and sends it to the e-mail address as shown in the "rua=" tag.

TABLE II shows the percentage of each DMARC policy based on the number of domains that we have observed. As shown in table, since the most of domains' DMARC policies

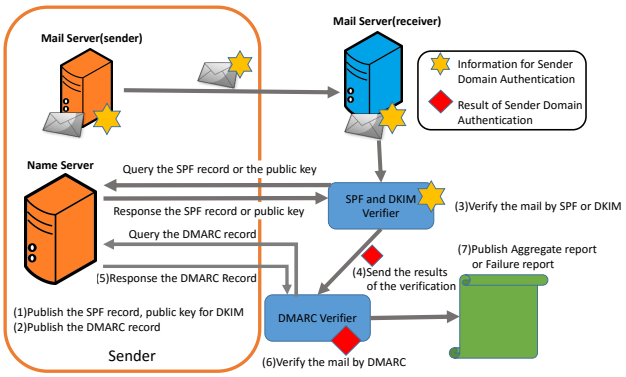

Fig. 2. Flow of DMARC Verification

TABLE II
SURVEY OF DMARC POLICY

Polocy	2016/2	2016/3	2016/4
none	1,473 (81.65%)	1,261 (82.31%)	1,821 (77.79%)
quarantine	123 (6.82%)	93 (6.07%)	209 (8.93%)
reject	192 (10.64%)	170 (11.10%)	305 (13.03%)
error	16 (0.89%)	8 (0.52%)	6 (0.26%)
Total	1,804 (100%)	1,532 (100%)	2,341 (100%)

are published as "none", the receiver will accept the verification-failed messages without rejection or quarantine. We can consider from this survey that many DMARC compliant sender domains hope receiver domains to accept spoofed messages as are and only to send aggregate reports. Hence the isolation or rejection effect of DMARC against spoofed e-mails is currently limited.

III. DESIGN OF DMARC VERIFICATION RESULT NOTIFICATION SYSTEM

A. *Summary of the System*

As described in Section II-A and Section II-B, DKIM cannot perform the verification for the e-mails that do not attach the digital signature. In other words, even if a received e-mail is from a domain that should have with a DKIM signature, DKIM cannot determine the e-mail that does not exist a DKIM signature as spoofed e-mail. To solve the problem, we propose a system to warn of such e-mails by utilizing DMARC. This system does not focus on creating and sending the reports explained in Section II-C.

Our proposed system performs sender domain authentication and notification of DMARC perform on users' terminals. By performing on users' terminals, PC users can easily adopt the sender domain authentication mechanisms and/or DMARC verification even if the user's mail receiving server does not support these mechanisms. The system obtains the mail receiving server information required for SPF from "Received" field of the mail header. After that, the system determines the boundary of the internal and external organization, and the system uses IP address of the nearest

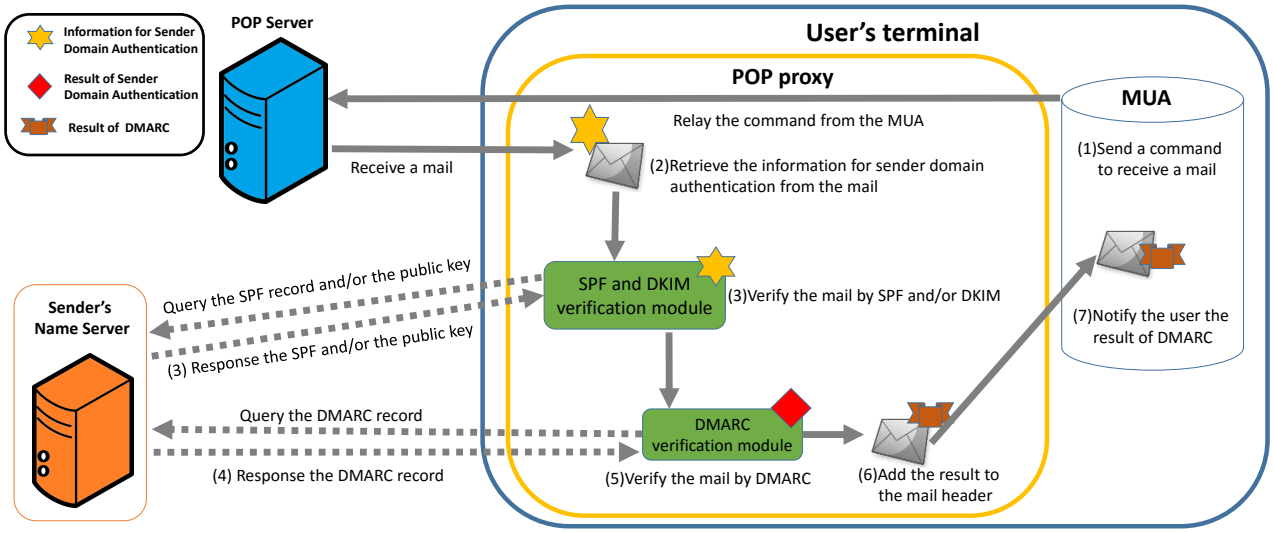

Fig. 3. Flow of DMARC Verification by Our System

external organization to the boundary and the e-mail address indicated by "Return-Path" for SPF verification.

SPF and DKIM verification are performed by the verification module shown in Fig.3. DMARC verification module receives the results of sender domain authentication and determines whether to apply the DMARC policy. DMARC verification module judges "pass" or "fail" as the verification result. Subsequently the system notifies the verification result to MUAs.

B. Summary of the System

Fig.3 shows the behavior of the POP proxy and a client in this system.

1) When the POP proxy received a message acquisition command from a MUA, the proxy relays the command to the POP server.
2) The proxy retrieves the information required for authentication from the header of the acquired e-mail, and inputs the information to the SPF and DKIM verification module.
3) SPF and DKIM verification module performs sender domain authentications based on the information obtained from the header.
4) DMARC verification module queries to the sender's DNS contents server, and acquires the DMARC record.
5) DMARC verification module applies the DMARC policy based on the result of the sender domain authentications.
6) The proxy adds the DMARC verification result to the mail header, and delivers the e-mail to the MUA.
7) The MUA notifies the user the DMARC verification result.

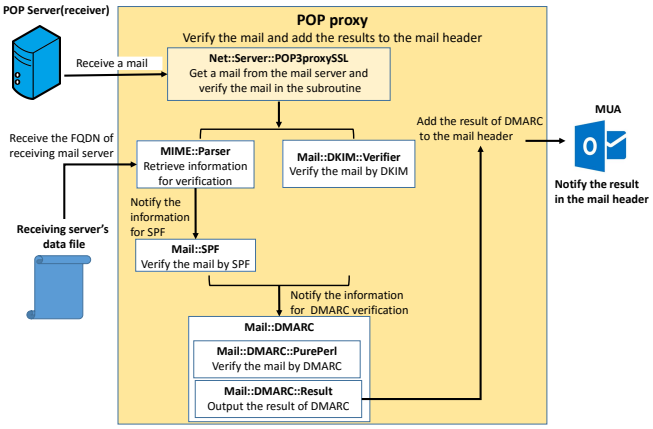

Fig. 4. Structure of Our System

IV. Implementation of the DMARC Verification Result Notification System

Based on the design described in Section II-C, we have developed the system using Perl. In order to perform DMARC verification, this system is configured by using Mail::DMARC and Mail::DMARC::PurePerl [6] that are modules published on CPAN. We used MIME::Parser [7] and Net::Server::POP3proxySSL, that was created based on Net::Server::POP3proxy, to obtain the information required for the verification from the mail header. In addition, by implementing them all on Cygwin, our proposed system works on a user's terminal.

First, we describe an implementation method of the part to obtain the information required for verification from a mail header. The parts necessary for verification are Return-Path, DKIM signature, From:, and To: in a mail header shown in Fig.1.

TABLE III
THE FIELDS THAT CAN BE OBTAINED BY MAIL::DMARC::RESULT

Field	Contents
result	DMARC verification result. (pass. fail)
disposition	DMARC policy when the result field is “fail”.
reason	The reason of the verification failure when the result field is “fail”.
dkim	The result of DKIM verification.
dkim_align	The degree of coincidence with the DKIM signature domain and the Header-From domain.
spf	The result of SPF verification.
spf_align	The degree of coincidence with the envelope-From domain and the Header-From domain.

Fig.4 shows the structure of our proposed system. The operation of POP proxy in this system can be divided into five of 1) obtaining a message from the POP server, 2) analysis of the mail header, 3) execution of the sender domain authentication, 4) execution of DMARC verification, 5) addition of the verification result. We describe about the implementation method for each of these steps.

1) In order to implement the POP proxy and obtain a message, we used Net::Server::POP3proxySSL that was created based on Net::Server::POP3proxy. This module receives a message from the POP server and stores it in “$[0]”. By passing the variable to “flterAction” that is a subroutine function, this module can perform processing on the message.
2) We used the MIME::Parser for the header of the analysis. This module isolates the mail header and the body, and extracts the necessary information using regular expression. Additionally, the module retrieves the sender information to be used for SPF verification from the “Received” field. The sender's information used for SPF verification is indicated on “Received” field the server located in the boundary of the internal and external organization is added to the header. Then, the system reads the receiving server's data file that retains the information of own organization's receiving server, and scans “Received” field. By preparing the external file, each organization is able to specify the receiving server without modifying the program code. When the appropriate “Received” field is specified and the source IP address is obtained, the module terminates.
3) The system performs SPF verification by using the information that was extracted with 2). We utilized Perl module Mail::SPF [8] for SPF verification. The system performs SPF verification by passing the sender IP address and Envelope-From address to this module. On the other hand, DKIM need to use the entire message for the verification. By passing “$_[0]”, that contains the entire message, to Perl module Mail::DKIM::Verifier [9], the system performs DKIM verification.
4) By using the information extracted in 2) and the result of authentication performing in 3), the system performs DMARC verification by Mail::DMARC::PurePerl which is a method of Perl module Mail::DMARC. The system performs the verification by passing the sender IP address, Envelope-From address, Header-From address, and verification results of SPF and DKIM to Mail::DMARC::PurePerl.
5) The system appends the DMARC verification result obtained in 4) to the mail header, and delivers to the MUA. The system receives DMARC verification result from DMARC::Mail::Result method. TABLE III shows the fields about verification results that are possible to obtain by this method. The system adds the verification result and Header-From domain regardless of the verification result to the mail header. Moreover, when the verification result is “fail”, this method can obtain the failure reason from “reason” field. Therefore even though the verification result was “fail”, this system can obtain “no_policy” as the failure reason from the “reason” field when the sender domain was not supported DMARC. In addition, as described in Section II-C, DMARC is different from DKIM, the verification will be failed when the domain indicated by the “d=” tag and the domain indicated by the Envelope-From address are different.
   The “spf_align” and “dkim_align” field indicates “strict” when the header from domain and the domain for each verification are completely consistent, and when each of these domains is the relationship of the subdomains, the field indicates “relaxed”. On the other hand, these fields do not have information when DMARC verification failed due to these domains are different.
   The system appends the domain of the receiver's e-mail address, SPF verification result and Envelope-From, DKIM verification result and its signed domain, DMARC verification result, and Header-From domain to the mail header. Additionally, when the verification fails, the system appends the reason. Therefore RFC7601[10] allows the freely description in the parentheses, the system adds the DMARC policy and the reason of verification failure as shown in Fig.5.

```
Authentication-Results: user.example.com;
  spf=pass smtp.mailfrom=sender@example.net;
  dkim=pass header.d=example.net;
  dmarc=fail (p=reject comment=no match)
  header.from=sender@example.com
```

Fig. 5. Addition of the Verification Result to the Mail Header

V. NOTIFICATION OF DMARC VERIFICATION RESULT

We implemented the notification function of DMARC verification results by using the label of Microsoft Outlook 2013 as a user's MUA. The system notifies the four types shown in the lower part of the Fig.6 based on DMARC

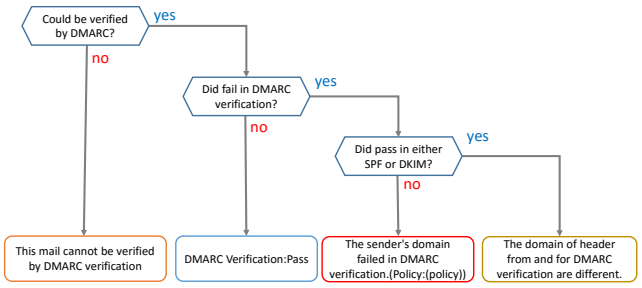

Fig. 6. Flow of the Label Addition

verification result.

Moreover, Fig.7-10 show the actual label additional examples in the MUA. The system appends a blue label when succeeding in the verification (Fig.7), and the system adds a yellow label when the verification failed and the sender domain indicated "none" or "quarantine" as the DMARC policy (Fig.8). Furthermore, the system appends an orange label when the sender does not correspond to DMARC (Fig.9), and the system adds a red label when the verification failed and the sender domain indicated "reject" as the DMARC policy (Fig.10).

Additionally, when the applied policy was "reject", that represents such e-mails did not attach the DKIM signature even though all of the legitimate transmissions that send from the domain are supposed to be attached the signature. Otherwise, such e-mails mean that failed to the verification. In any case such mails are extremely high possibility of spoofing or falsification, therefore the system alerts by pop-up window in addition to the red label notification as shown in Fig.11.

VI. DISCUSSION

In general usage of DMARC, a receiver does not handle spoofed e-mails unless the sender's DMARC policy is "reject" or "quarantine". However, as shown in TABLE II, about 80% of the DMARC compliant domains publish "none" as the policy. Therefore, the existing systems cannot isolate or reject the e-mails even if those are very high probability of being spam mails. On the other hand, by giving various warnings according to each policy, our system enables alerting to spoofed e-mails that the conventional systems cannot warn.

Moreover, since DMARC can be expected to spread more widely in the future, the effectiveness of this system will be increased.

VII. CONCLUSION

In this paper, we proposed a system that distinguishes spoofed e-mails utilizing DMARC. Our proposed system can alert spoofed e-mails that do not attach the DKIM signature even though all of the legitimate transmissions that send from the domain are supposed to be attached the signature. A

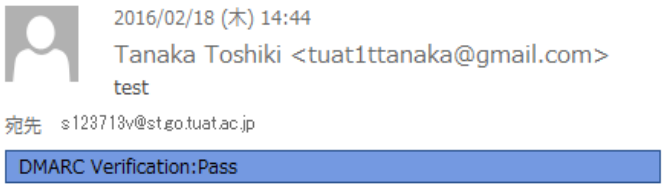

Fig. 7. Addition of "pass" Label

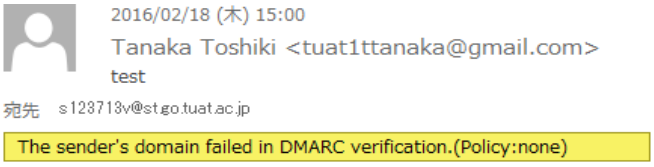

Fig. 8. Addition of "none" Label

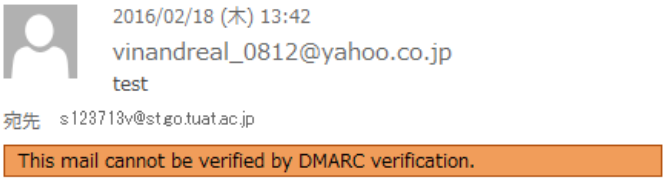

Fig. 9. Addition of "Non-DMARC-compliant" Label

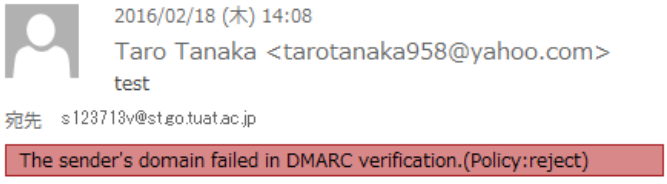

Fig. 10. Addition of "reject" Label

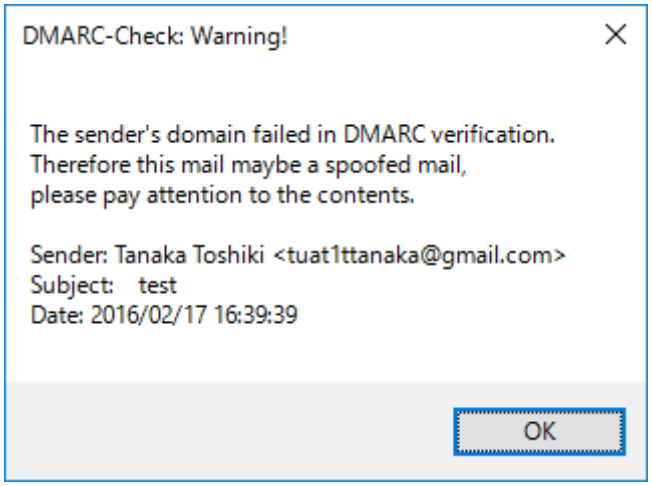

Fig. 11. Pop-up Alert Window

remarkable point of the system is to implement the all functions of sender domain authentication, DMARC verification, and the result notification on a user's PC. By implementing on each user's PC, users can install a spoofed e-mail alert system even if their receiving server does not support DMARC verification. Generally DMARC is used for administrators of sender domain receives the report of sender domain authentication. However, this system is able to alert the spoofed e-mails by visually notifying DMARC verification result to each recipient. Moreover, even when the sender domain was publishing "none" as the DMARC policy, our system can prevent a recipient overlooking the spoofed e-mails by the notification.

This system performs sender domain authentication and DMARC verification in POP proxy, thus the system is only compatible with POP. Therefore e-mail receiving via IMAP has been widely utilized in recent years, support of the mechanism described in this paper to the IMAP environment is a future subject.

ACKNOWLEDGMENT

We would like to thank Mr. Ayachika Kitazaki, who is the vice chairman of anti spam committee of The Internet Association Japan, for providing us data of DMARC policy statistics.

REFERENCES

[1] FBI (Federal Bureau of Investigation): Public Service Announcement, E-mail Account Compromise. [Online]. Available: http://www.ic3.gov/media/2015/150827-2.aspx
[2] M. Wong and W. Schlitt. (2006, Apr.). Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, RFC4408, IETF
[3] D. Crocker, T.Hansen, M. Kucherawy. (2011, Sep.). DomainKeys Identified Mail (DKIM) Signatures, STD76, IETF
[4] M. Kucherawy, E. Zwicky. (2015, Mar.). Domain-based Message Authentication, Reporting, and Conformance (DMARC), RFC7489, IETF
[5] M. Fukuyama, M. Oiwa, N. Yamai, N. Kitagawa, "Implementation of DKIM Verification System Using POP Proxy," IPSJ Technical Report, 2015-IOT-28, No.2, 2015, pp.1-6 (in Japanese).
[6] CPAN:MAIL::DMARC.pm. [Online]. Available: http://search.cpan.org/~msimerson/Mail-DMARC-1.20150527/lib/Mail/DMARC.pm
[7] CPAN:MIME::Parser.pm. [Online]. Available: http://search.cpan.org/~dskoll/MIME-tools-5.506/lib/MIME/Parser.pm
[8] CPAN:Mail::SPF.pm. [Online]. Available: http://search.cpan.org/~jmehnle/Mail-SPF-v2.9.0/lib/Mail/SPF.pm
[9] CPAN:Mail::DKIM::Verier.pm. [Online]. Available: http://search.cpan.org/~jaslong/Mail-DKIM/lib/Mail/DKIM/Verifier.pm
[10] M. Kucherawy. (2015, Aug.). Message Header Field for Indicating Message Authentication Status, RFC7601, IETF

Naoya Kitagawa received his B.Sc. and M.Sc. degree in information science from Chukyo University, Toyota, Japan, in 2009 and 2011 respectively, and his Ph.D. degree in information science from Nagoya University, Nagoya, Japan, in 2014.
In April 2014, he joined Information Technology Center, Nagoya University as a postdoctoral fellow. Since October 2014, he has been an assistant professor in the Institute of Engineering, Tokyo University of Agriculture and Technology. His research interests include the Internet, network security, and distributed system. He is a member of IPSJ.

Toshiki Tanaka received B.E. in computer and information science from Tokyo University of Agriculture and Technology, in 2016.

Masami Fukuyama received B.E. in computer and information science from Tokyo University of Agriculture and Technology, in 2016. Since April 2016, he has been a graduate student in the Graduate School of Engineering, Tokyo University of Agriculture and Technology.

Nariyoshi Yamai received his B.E. and M.E. degrees in electronic engineering and his Ph.D. degree in information and computer science from Osaka University, Osaka, Japan, in 1984, 1986 and 1993, respectively.
In April 1988, he joined the Department of Information Engineering, Nara National College of Technology, as a research associate. From April 1990 to March 1994, he was an Assistant Professor in the same department. In April 1994, he joined the Education Center for Information Processing, Osaka University, as a research associate. In April 1995, he joined the Computation Center, Osaka University, as an assistant professor. From November 1997 to March 2006, he joined the Computer Center, Okayama University, as an associate professor. From April 2006 to March 2014, he was a professor in the Information Technology Center (at present, the Center for Information Technology and Management), Okayama University. Since April 2014, he has been a professor in the Institute of Engineering, Tokyo University of Agriculture and Technology. His research interests include distributed system, network architecture and Internet. He is a member of IEICE, IPSJ and IEEE