J Inf Process Syst, Vol.19, No.3, pp.334~346, June 2023
https://doi.org/10.3745/JIPS.04.0276

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Design and Implementation of the Document HTML System for Preserving Content Integrity

Hyun Cheon Hwang[1], Ji Su Park[2], and Jin Gon Shon[3,*]

Abstract

An electronic document based on PDF has been widely used in customer communication between an enterprise and a customer to deliver personalized content. However, electronic documents based on PDF in the form of paper layouts are not suitable for mobile environments because of low readability and lack of interactive interaction. Even though HTML is an essential language in a mobile environment, electronic document based on PDF is still used as it has a content integrity verification feature with a digital signature. It means that a user is sacrificing user experience in a mobile environment for content integrity and using paper-layout electronic documents. In this research, we design the Document HTML specification by setting the Document HTML conformance, adding the extended meta tags, and signing the message digest with a digital signature based on public key infrastructure (PKI). Furthermore, we implemented the Document HTML system, which has REST API services to generate and verify the Document HTML, and did experimental verification of the theory. As a result, we have confirmed that the Document HTML has both content integrity and user experience on mobile. Furthermore, the Document HTML is expected to be an alternative document format to deliver personalized content from an enterprise to a customer in a mobile environment instead of the paper layout electronic document such as PDF.

Keywords

Content Integrity, Customer Communication, Digital Signature, Document HTML, HTML

1. Introduction

An enterprise such as a financial company sends personalized content to a customer to engage with them. The interaction between an enterprise and a customer is called customer communication. In the past, most customer communication was done by a postal letter, a call, or a physical contact. Furthermore, customer communication has been moved on digital along with emerging of a computer. Moreover, recent customer communication has been moved from PC to mobile with the explosive growth of a smartphone [1]. A traditional electronic document with a paper layout is used on a PC, which has a bigger screen than a standard paper, could not be a relevant document on mobile, which has a small screen size, because of readability and some lack of interactive features. Although an HTML document is the most proper document format on mobile, this is not the suitable format to deliver sensitive personal customer data because an hypertext markup language (HTML) document is not a single file to deliver content and

Manuscript received January 21, 2022; first revision March 10, 2022; accepted March 18, 2022.
* **Corresponding Author:** Jin Gon Shon (jgshon@knou.ac.kr)
[1] Dept. of Industrial and Information System, Graduate School, Seoul National University of Science and Technology (a.hwang@seoultech.ac.kr)
[2] Dept. of Computer Science and Engineering, Jeonju University (jisupark@jj.ac.kr)
[3] Dept. of Computer Science, Graduate School, Korea National Open University (jgshon@knou.ac.kr)

an HTML document combines all external resources to display content on the fly. Moreover, there is no content integrity verification feature, and as a result, an HTML document cannot be used as evidence in case legal disputes happen. Alternately, a portable document format (PDF) document is still selected because the content integrity feature of a PDF is essential for the document, which needs to deliver sensitive personal information. However, it gives a bad user experience as a PDF document is specialized for paper layout formatting, and a customer may abandon their expectation for the user experience on mobile [2]. Because of these limitations of HTML and PDF documents, a suitable document format on mobile to have a better user experience and content integrity is required. In this paper, we research the Document HTML format by designing the Document HTML conformance and public key infrastructure (PKI) digital signature, which can provide content integrity as well as better user experience such as responsive layout for different devices.

We had related research for the Document HTML in Section 2, designed the Document HTML in Section 3, and did experimental verification through the Document HTML system and a sample HTML document in Section 4. In last, we had the conclusion of the research in Section 5.

2. Related Research

2.1 Electronic Document

The definition of a document can be described as a unit of "recorded information structured for human consumption" [3] and this definition also accommodates a wide variety of documents used in an organization such as contracts and agreements [3]. In Korean law, the definition of an electronic document is the information that is sent, received, or archived, which is created or converted electronically by the information processing system [4]. As we know through these definitions, a wide variety of documents can be converted to electronic documents format for human consumption on a digital device such as a PC. There are two types of electronic document formats based on a consumer perspective. A document format such as an extensible markup language (XML) is used to exchange information between information systems, and this format is mainly for consumption by a machine. In contrast, a PDF is used to present as a traditional paper document, and this format is mainly for consumption by a human. Even though other electronic document formats, such as DOCX, represent a paper layout, these formats are not independent of document resources and are vulnerable to content integrity. So, PDF is widely used as a de-facto standard document format in electronic document distribution. Even though the PDF is an independent document format against an operating system and a device, PDF is a fixed-layout document format, and PDF does not provide an enhanced user experience in a mobile environment, unlike HTML. Therefore, an HTML document is getting more major document format along with the mobile era even though HTML format has a weakness in content integrity rather than PDF.

2.2 PDF vs. HTML

Adobe developed PDF specification [5] and has become the ISO 32000 standard specification. A PDF document provides a paper layout experience on a digital device. It is the independent format against an operating system and a device as all resources, such as a font, can be embedded in the PDF file. So, a PDF document has the same layout on any digital device and any OS, whereas other document formats

do not provide the same paper layout. As a result, a PDF document has become a de-facto standard document format. The PDF specification also has a digital signature using a PKI certificate to verify the content integrity [6]. A creator creates a PDF document and adds a digital signature. A PDF viewer software can verify the content integrity whether the PDF document has not been altered after digital signing. By these characteristics, a PDF document is widely used in document communication, such as sending a legal contract from an enterprise to an end-user.

HTML is the markup language for web pages, and the latest version is HTML5.2 [7]. An HTML document consists of a main HTML and external resources such as cascading style sheet (CSS) and JavaScript. The dynamic web using HTML and external resources provides a responsive web customer experience. It is becoming more critical as the digital environment moves from a PC to a mobile environment. As a PDF has a weakness in terms of readability and responsive interaction in a mobile environment, an HTML document in electronic document territory is getting the next de-facto standard than a PDF document. However, an HTML document is a dependent document format that needs external resources to represent content, whereas a PDF is an independent format and can embed all resources internally. Moreover, an HTML document is in plain text format and easy to be altered by an unauthorized system, and there is no standard specification to secure the content integrity of the document. Because of this weakness, HTML document still has a vulnerability to being a trusted document used to be a standard document format in customer communication to deliver personalized content, including considering a legal dispute.

2.3 Content Integrity Verification

It is the most efficient way to verify digital content integrity by using the one-way hash algorithm. The hash function creates the fixed length's checksum data regardless of the size of the original content, and each checksum data has a unique value [8]. There are message-digest algorithm (MD) and secure hash algorithm (SHA) in the cryptography hash algorithm [9]. The main key feature in the hash algorithm is to have a unique output called collision avoidance. Hash algorithms are constantly being improved to avoid a collision or unauthorized attack. For instance, the collision has been found in the SHA-0 algorithm [10]. So, SHA has been improved, and SHA-3 has been published through SHA-1, SHA-2. SHA-3-512 algorithm creates a 512 bits length unique value for all different data inputs.

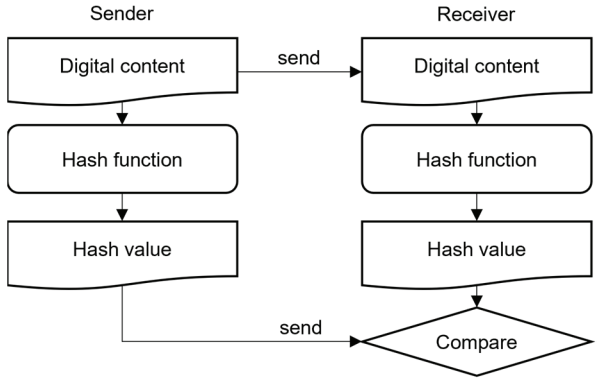

Fig. 1. Content integrity verification process.

As every digital content has a unique value by the hash algorithm, the digital content can be stored together with the hash value for content integrity verification. A reviewer who wants to verify the content integrity in the future can generate the hash value by using the same algorithm and comparing the original stored hash value. If the two values are identical, the content has not altered since it was created. A sender sends the digital content with the hash value together to a receiver. The receiver generates the hash value again to compare the received value, as shown in Fig. 1. However, there is no identity verification and no way to prove the received hash value is not manipulated. So, it has a security vulnerability because an unauthorized user could hijack the sender's digital content and the hash value. The manipulated digital content and the hash value could be delivered with malicious intent.

2.4 Digital Signature

A content integrity verification process based on the hash algorithm is the most efficient way. It creates a fixed-length unique value and a receiver re-generates a hash value to compare it. However, there is no identity verification of a sender, and anyone can create a hash value using a hash algorithm. RSA is an asymmetric cryptography algorithm consisting of public and private keys, as shown in Fig. 2 [11]. RSA algorithm is based on factoring by two randomly selected prime numbers. In RSA, the bit length of N is the RSA algorithm length. It takes around 2,000 MIPS years to do computing in the case RSA-140 bits [12]. In the case RSA-1024 bits, it takes 49,000,000 times more than RSA-140. The recent RSA algorithm is used more than 2,048 bits, and the RSA algorithm is used widely for cryptography with its robust security mechanism.

$$
\begin{aligned}
&p, q = prime\ number; p \neq q \\
&N = pq \\
&\emptyset(N) = (p-1)(q-1) \\
&\gcd(\emptyset(N), e) = 1; 1 < e < \emptyset(N) \\
&de \bmod \emptyset(N) = 1 \\
&public\ key = \{N, e\} \\
&private\ key = \{N, d\}
\end{aligned}
$$

Fig. 2. RSA algorithm.

A PKI is a set of required procedures to create, verify, manage and other actions for a digital signature [13]. In PKI, a trusted certificate authority (CA) creates a set of a public key and a private key using a cryptography algorithm such as RSA. A signer can get the set of a public key and a private key with the CA information. The signer encrypts the message digest created from the digital content using a private key and sends the encrypted message digest, a public key and CA information. The receiver decrypts the encrypted message digest using a public key and can compare it with the received message digest to verify the content integrity as shown in Fig. 3. As a receiver does not have a private key, a receiver can only do the decryption. In addition, the receiver can validate that the public key is created by a trusted certificate authority through the CA information.

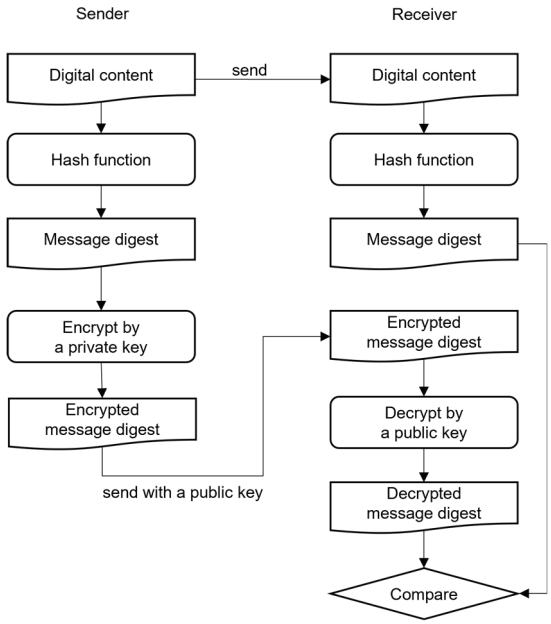

Fig. 3. Digital signature process.

3. Design of the Document HTML System

3.1 Document HTML Conformance

An HTML document consists of multiple files, and it is hard to manage the content integrity as there is no strong cohesiveness among files. The Web ARChive (WARC) format is the standard ISO 28500 for archiving web contents, including an HTML document. It generates a single WARC file containing HTML and related resources [14]. And the MHTML (MIME HTML) format based on RFC 2557 generates a single MHTML file that contains an HTML and related resources [15]. However, the WARC file is not the standard file format that can be opened on a web browser, so an end-user cannot open this file. In addition, there is a standard specification for all types of web browsers to display an MHTML file so that MHTML can be displayed differently on a different web browser. Moreover, both WARC and MHTML specifications do not have a standard specification to verify content integrity, so it has a vulnerability.

The Document HTML is the specification based on HTML to be treated as a document. The Document HTML conformance does not allow an external resource. All resources must be embedded inside the Document HTML for the integrity of the resources to avoid being shown different content due to updating an external resource. Furthermore, the Document HTML needs to have a verification feature of the content integrity if the unauthorized way alters it. Hence, the Document HTML conformance requirements are as shown in Table 1.

Table 1. Requirements for the Document HTML

Requirements
R1. All related resources must be in an HTML document internally.
R2. An HTML document must be signed by a digital signature.

First, all related resources must be in an HTML document internally to be treated as a single document, as shown in R1 of Table 1. Furthermore, the Document HTML does not allow the specific HTML tags, which can open another HTML page inside the Document HTML, so the <iframe> tag is not allowed to be used in the Document HTML. Furthermore, to load content using async data loading technology such as AJAX is not recommended as it can display different content from a server-side even though the Document HTML is identical. Second, the Document HTML must be signed by a digital signature as shown in R2 of Table 1. The Document HTML is the single file after embedding all resources, and it can be digital signed using PKI. Hence, the Document HTML can remove the vulnerability in terms of content integrity, and the Document HTML is suitable for delivering personalized content.

3.2 An Embedded Resource for the Document HTML

The Document HTML uses the Data URL scheme to embed a resource internally, as shown in Fig. 4. The Data URI is the standard specification as RFC 2397 [16], and a resource can be embedded in BASE64 format with a prefix data type indicator. For instance, an image file representing a logo in an HTML document can be embedded with an image file format indicator, as shown in Fig. 5. When creating the Document HTML, the Document HTML system converts an external resource to an embedded resource.

```
data:[<mediatype>][;base64],<data>
```

Fig. 4. Data URI scheme.

```
<html>
  <head/>
  <body>
      <img src=" iVBORw0KGgoAA.. "/>
  </body>
</html>
```

Fig. 5. Example of the Document HTML with an embedded image.

3.3 The Document HTML Meta Tags

The HTML document follows the rule: all related resources must be in an HTML document internally, as shown in R1 of Table 1 can be a single file. And the HTML document can present content without loading resources from outside of the HTML document. However, there is a vulnerability regarding content integrity as there is no content integrity verification feature. We define the extended HTML meta tags for the Document HTML to include a digital-signed value, as shown in Fig. 6 to have a content integrity verification feature. The message digest is generated by a hash function such as SHA, and this value is located in the ds-digest extended meta tag with hash function indicator, as shown in Fig. 7. A receiver can acknowledge which hash function is used for the message digest by hash function indicator. The message digest in ds-digest is signed by a private key from a digital certificate, and the signed value is located in the ds-signed-digest extended meta tag. Last, the public key and certification information value is located in the ds-cert extended meta tag. The byte position of the target content area in the HTML for digital signing is located in the ds-range extended meta tag, as shown in Fig. 8. The begin and end byte position of the above area of the extended meta tags and the begin and end byte position of the below

area of the extended meta tags are in the ds-range. The ds-range has four-position values as a hexadecimal expression to save space. The four-position values are fixed-length values to avoid misdirecting the below area position, and each value can be from 0x00000000 to 0xFFFFFFFF as shown in Table 2. The maximum file size with the ds-range is 4,294,967,295 bytes as the maximum position value is 0xFFFFFFFF. This maximum size is sufficient to handle the Document HTML to deliver personalized content. The Document HTML specification from the previous research used an HTML comment tag to include digital signature values [17]. It is not easy to access these values with an HTML feature. Therefore, we define the extended HTML meta tag by HTML meta tag, and it provides better accessibility with a standard HTML parsing library.

```
<meta name="ds-range" content="Byte Position"/>
<meta name="ds-digest" content="Labeled Message Digest"/>
<meta name="ds-signed-digest" content="Signed Message Digest"/>
<meta name="ds-cert" content="Certificate"/>
```

Fig. 6. Extended meta tag in the Document HTML

```
<meta name="ds-digest" content="sha256:pmWkWSBCL51Bfkh"/>
```

Fig. 7. An example of the ds-digest meta tag.

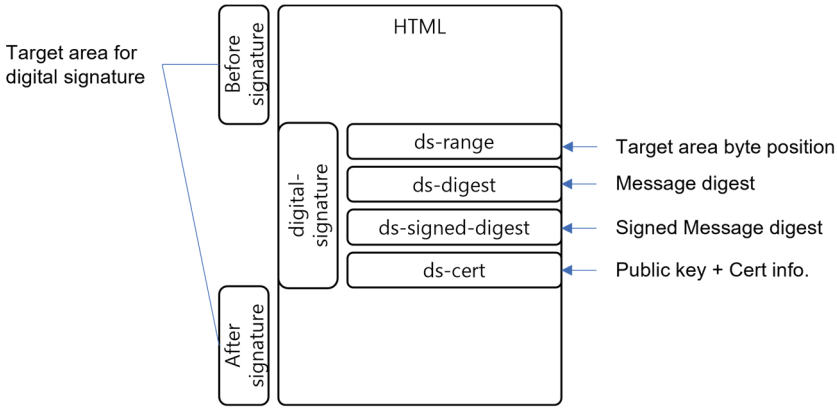

Fig. 8. Document HTML structure with the digital signature.

Table 2. Extended meta tag in the Document HTML

tag	"ds-range"
content	"Byte Position"
content format	"[0-9a-f]{8} [0-9a-f]{8} [0-9a-f]{8} [0-9a-f]{8}"
example	<meta name="ds-range" content="00000000 000000FF 00000AFF 00000BFF"/>

3.4 Document HTML Generation and Verification Process

The Document HTML system has the Document HTML generation process as shown in Fig. 9. An HTML document that is created in a legacy system will be converted to the Document HTML to have content integrity using the Document HTML system before delivering it to an end-user. The Document

HTML generation process is to receive an HTML document and convert external resources to internal resources, and the message digest of the HTML document is generated. After that, the message digest is signed by a private key. The signed message digest, the public key, and the certification information will be added to the HTML document via the extended meta tags. Eventually, the Document HTML is returned to the legacy system.

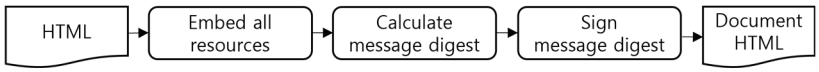

Fig. 9. The Document HTML generation process.

The Document HTML contains all related resources and the digital certificate, and it can be verified content integrity as shown in Fig. 10. The validation process is to receive the Document HTML, parse the Document HTML meta tags, generate the message digest from the Document HTML, and get the delivered message digest from the signed message digest using a public key. Last, the two message digests are compared to check the content integrity.

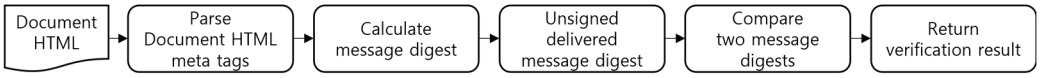

Fig. 10. The Document HTML verification process.

4. Experimental Verifications of the Document HTML System

4.1 Implementation Environment

The Document HTML system needs to have two entry points for the Document HTML generation and verification processes. The generation process needs to communicate with a sender who creates the Document HTML, such as an enterprise legacy system. The verification process needs to communicate with a receiver who wants to verify the Document HTML, such as an end-user. As the Document HTML system can be implanted in the cloud environment, we define the API by RAML (RESTful API Modeling Languages) [18]. The Document HTML system provides two major REST APIs to generate and verify the Document HTML as shown in Table 3.

The Document HTML system is implemented by Django web framework on Linux OS, and the Document HTML system used the Let's Encrypt certificate for digital signature. OpenSSL was used to sign a message digest and verify a signed message digest.

4.2 The Document HTML Generation

We prepared the sample HTML document to simulate a billing statement from an enterprise based on the research. The sample HTML document has the logo image, CSS style file, and JavaScript library as external resources. The sample HTML document represents a personal billing address, an amount to pay, and detailed transactions like a billing statement. The sample HTML document was sent to the Document HTML generation process via REST API and the Document HTML was created as shown in Fig. 11.

Table 3. REST APIs of the HTML Document system

API	Generate the Document HTML	Verify the Document HTML
API entry point	/api/document-html/generate	/api/document-html/verify
Method	POST	POST
Request type	application/json	application/json

Request body	{
	"AuthKey": Auth API Key,
	"data": HTML Document
	}
		{
		"AuthKey": Auth API Key,
		"content": the Document HTML
		}

Response type	text/html	application/json

Response body	The Document HTML	{
		"is_valid": True or False,
		"content_digest": Content message digest,
		"message": Validation message,
		"cert_info": Certificate information
		}

Fig. 11. The generated Document HTML from the sample HTML.

The Document HTML includes the relative resources as an internal format using Data URI as shown in ⑤ of Fig. 11. The ds-range meta tag contains byte positions to indicate the target content area in the Document HTML, as shown in ① of Fig. 11. The message digest is calculated and is located in the ds-digest meta tag, as shown in ② of Fig. 11. The signed message digest which is created using the private

key is in the ds-signed-digest meta tag, and the certificate information and the public key are in the ds-cert meta tag to verify content integrity, as shown in ③ and ④ of Fig. 11.

The file size of the Document HTML is increased as there are additional extended meta tags and converting internal resources as base64 format, as shown in Fig. 12.

$$fs(d) = fs(h) + fs(m) + \sum_{i=1}^{n} fs(fb(r_i))$$

$fs = file\ size$
$d = the\ Document\ HTML$
$m = meta\ tags$
$n = the\ number\ of\ resources\ in\ the\ Document\ HTML$
$fb = base64$
$h = HTML$
$r = external\ resources\ in\ HTML$

Fig. 12. The size of the Document HTML.

4.3 The Document HTML Verification

For the experimental verification, we considered and made the possible scenarios that can be happened, as shown in Table 4. Scenario 1 is the verification of the undamaged Document HTML, so the verification result, which is the valid Document HTML, is expected. Scenario 2 shows that an unauthorized user modifies a content to display different content, such as scamming, so the verification result, which is the invalid Document HTML, is expected. Scenarios 3–5 show that an unauthorized user modifies the digital signature metadata area, so the verification result, which is the invalid Document HTML, is also expected.

Table 4. Scenarios for the Document HTML verification experiment

	Description	Expected result
Scenario 1	Undamaged	Valid Document HTML
Scenario 2	Damaged in a content	Invalid Document HTML
Scenario 3	Damaged in a message digest	Invalid Document HTML
Scenario 4	Damaged in a singed message digest	Invalid Document HTML
Scenario 5	Damaged in a certificate information	Invalid Document HTML

All Document HTML was sent to the REST API verification process, and the response via REST API can be displayed as shown in Fig. 13. The verification result of the valid Document HTML showed the Document HTML has content integrity and the document is not damaged, as shown in Fig. 13(a). On the other hand, the verification result of the damaged Document HTML showed the document is damaged and this is not a trusted document, as shown in Fig. 13(b).

Even though there is the file size overhead, the Document HTML is a single file including the digital signature so that to have content integrity. We used the digital certificate from Let's Encrypt, which has a 2,048 bits key length. As the Document HTML uses the PKI mechanism, it will have the same level of security against unauthorized modification manner. It takes around 2,000 MIPS years to do computing in case RSA-140 bits and it takes 49,000,000 times more than RSA-140 with RSA-1024 bits [12]. The digital signature we used in the research has a 2,048 bits length. Hence, it can be said that it is impossible to alter the Document HTML file in an unauthorized way. It provides strong content integrity and a mobile-friendly user experience.

(a) (b)

Fig. 13. The verification results: (a) the valid Document HTML and (b) the invalid Document HTML.

As a result, we have confirmed the Document HTML provides content integrity like a traditional electronic document based on PDF as well as a mobile-friendly user experience as the Document HTML is created based on HTML. The Document HTML is the suitable electronic document format to deliver the content which contains legalistic personalized content in the enterprise such as financial service industry (FSI) or e-Government as the Document HTML has content integrity verification and can be long-term archived on the mobile environment.

5. Conclusion

In this paper, we have proposed the Document HTML specification and implemented the Document HTML system to make an electronic document based on HTML a trusted electronic document as well as a user-friendly electronic document on mobile. In addition, we did experimental verification of the Document HTML system with a sample billing statement HTML document. We have confirmed that the Document HTML provides a content integrity verification feature. The Document HTML is suitable to deliver personalized sensitive content that needs to be verified or long-term archived. An electronic document, which has a physical paper layout such as PDF, is widely used on behalf of a physical letter to communicate between an enterprise and an end-user. However, this type of electronic document does not provide a good user experience on mobile, and it is hard to do bi-directional communication. In contrast, this electronic document provides content integrity verification. Hence, Document HTML is expected to be an alternative next electronic document format on mobile to have a better customer experience and content integrity verification. An enterprise should deliver sensitive personalized content, including legal information that can efficiently communicate with an end-user using the Document HTML on mobile. However, the Document HTML system is managed centrally, and it has a weakness of centralized management. In future research, we will research the de-centralized Document HTML system.

References

[1] Cisco, "Cisco Annual Internet Report," 2021 [Online]. Available: https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/index.html.

[2] Z. Shelton and C. H. Yu, "PDF readability enhancement on mobile devices," in *Proceedings of the 17th International Web for All Conference*, Taipei, Taiwan, 2020, pp. 1-4.

[3] R. H. Sprague, "Electronic document management: challenges and opportunities for information systems managers," *MIS Quarterly*, vol. 19, no. 1, pp. 29-49, 1995.

[4] H. C. Kim, "Issues and subjects of the framework act on electronic document and electronic commerce," *Law Studies*, vol. 15, no. 2, pp. 293-322, 2012.

[5] J. E. Warnock and C. Geschke, "Founding and growing Adobe Systems, Inc.," *IEEE Annals of the History of Computing*, vol. 41, no. 3, pp. 24-34, 2019.

[6] S. Rohlmann, V. Mladenov, C. Mainka, and J. Schwenk, "Breaking the specification: PDF certification," in *Proceedings of 2021 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, 2021, pp. 1485-1501.

[7] W3C, "HTML 5.2," 2023 [Online]. Available: https://www.w3.org/TR/html52/.

[8] G. Sekar and S. Bhattacharya, "Practical (second) preimage attacks on the TCS_SHA-3 family of cryptographic hash functions," *Journal of Information Processing Systems*, vol. 12, no. 2, pp. 310-312, 2016.

[9] H. Gilbert and H. Handschuh, "Security analysis of SHA-256 and sisters," in *Selected Areas in Cryptography*. Heidelberg, Germany: Springer, 2013, pp. 175-193.

[10] S. Manuel and T. Peyrin, "Collisions on SHA-0 in one hour," in *Fast Software Encryption*. Heidelberg, Germany: Springer, 2008, pp. 16-35.

[11] J. H. Song, S. S. Kim, and M. S. Jun, "Diffie-Hellman based asymmetric key exchange method using collision of exponential subgroups," *KIPS Transactions on Software and Data Engineering*, vol. 9, no. 2, pp. 39-44, 2020.

[12] S. Contini, "The factorization of RSA-140," *RSA Laboratories' Bulletin*, vol. 1999, no. 10, pp. 1-2. 1999.

[13] R. Perlman, "An overview of PKI trust models," *IEEE Network*, vol. 13, no. 6, pp. 38-43, 1999.

[14] M. Toyoda and M. Kitsuregawa, "The history of web archiving," *Proceedings of the IEEE*, vol. 100(Special Centennial Issue), pp. 1441-1443, 2012.

[15] J. Palme and A. Hopmann, "MIME E-mail encapsulation of aggregate documents, such as HTML (MHTML)," Internet Engineering Task Force, Fremont, CA, *RFC2110*, 1997.

[16] L. Masinter, "The 'data' URL scheme," Internet Engineering Task Force, Fremont, CA, *RFC 2397*, 1998.

[17] H. C. Hwang and W. J. Kim, "Design of document-HTML generation technique for authorized electronic document communication," *Journal of Society of Korea Industrial and Systems Engineering*, vol. 44, no. 1, pp. 51-59, 2021.

[18] B. Choi, J. Lee, S. Park, and J. Lee, "A model-based interface to cloud services for intelligent service robots," *KIPS Transactions on Software and Data Engineering*, vol. 9, no. 1, pp. 1-10, 2020.

Hyun Cheon Hwang **https://orcid.org/0000-0003-3841-5570**

He received B.S. degree in Computer Science from Dongguk University in 2001 and received M.S. degree in Computer Science from Korea Open National University in 2016, respectively. Ph.D. degree in Industrial and Information system from Seoul National University of Science and Technology in 2022, and he is currently the Principal Solution Architect in Quadient. His research interests are in Omni-channel communication, Digital archiving, and Blockchain.

Ji Su Park **https://orcid.org/0000-0001-9003-1131**

He received his B.S. and M.S. degrees in Computer Science from Korea National Open University, Korea, in 2003 and 2005, respectively and Ph.D. degrees in Computer Science Education from Korea University, 2013. He is currently a Professor in Dept. of Computer Science and Engineering from Jeonju University in Korea. His research interests are in mobile grid computing, mobile cloud computing, cloud computing, distributed system, computer education, and IoT. He is employed as editor-in-chief of *KIPS Transactions on Software and Data Engineering* by KIPS and managing editor & associate editor of *Human-centric Computing and Information Sciences* (HCIS) by Springer, and *The Journal of Information Processing Systems* (JIPS). He has received "best paper" awards from the CSA2018, BIC2021, CSA2021 conferences and "outstanding service" awards from CUTE2019 and BIC2020. He has also served as the chair, program committee chair or organizing committee chair at several international conferences including World IT Congress, MUE, FutureTech, CSA, CUTE, BIC.

Jin Gon Shon **https://orcid.org/0000-0002-0540-4640**

He received the B.Sc. degree in mathematics and the M.S. and Ph.D. degrees in computer science from Korea University, Seoul, Korea. Since 1991, he has been with the Department of Computer Science, Korea National Open University (KNOU). He had been a Visiting Professor for one year from August 1997 at State University of New York (SUNY) at Stony Brook, USA. After serving the Head of Information & Computer Center and the Head of e-Learning Center, Professor Shon had established the Department of e-Learning, the first master program of e-Learning in Korea, and served as the Chair of the Department until 2010. For 2 years after that, he had been working for KNOU as Director of the Digital Media Center, where all of KNOU e-learning contents and TV programs are produced. His research interests are in computer networks, distributed computing, and ITLET (Information Technology for Learning, Education, and Training) as a member of Korean Delegation to ISO/IEC JTC1/SC36 since 2000. He has made presentations in many conferences, and he won the Best Paper Award (Gold Medal) in the 24th AAOU Annual Conference in 2010. He has also published over 30 scholarly articles in the noted journals and written several books on computer science and e-learning.