J. Soc. Korea Ind. Syst. Eng Vol. 44, No. 1 : 51-59, March 2021
https://doi.org/10.11627/jkise.2021.44.1.051
ISSN : 2005-0461(print)
ISSN : 2287-7975(online)

Design of Document-HTML Generation Technique for Authorized Electronic Document Communication

Hyun-Cheon Hwang · Woo-Je Kim†

Public Policy and Information Technology, Seoul National University of Science & Technology

공인전자문서 소통을 위한 Document-HTML 문서 생성 기법의 설계

황현천 · 김우제†

서울과학기술대학교 IT정책대학원

Electronic document communication based on a digital channel is becoming increasingly important with the advent of the paperless age. The electronic document based on PDF format does not provide a powerful customer experience for a mobile device user despite replacing a paper document by providing the content integrity and the independence of various devices and software. On the other hand, the electronic document based on HTML5 format has weakness in the content integrity as there is no HTML5 specification for the content integrity despite its enhanced customer experience such as a responsive web technology for a mobile device user. In this paper, we design the Document-HTML, which provides the content integrity and the powerful customer experience by declaring the HTML5 constraint rules and the extended tags to contain the digital signature based on PKI. We analyze the existing electronic document that has been used in the major financial enterprise to develop a sample. We also verify the Document-HTML by experimenting with the sample of HTML electronic communication documents and analyze the PKI equation. The Document-HTML document can be used as an authorized electronic document communication and provide a powerful customer experience in the mobile environment between an enterprise and a user in the future.

Keywords : Document-HTML, Authorized Electronic Document Communication, HTML5, PKI

1. 서 론

페이퍼리스(Paperless) 시대가 열리면서 많은 종이 우편물이 디지털 우편물로 전환되고 있다. 이러한 디지털 우편물에는 기업과 개인 간의 법적 고지 및 계약의 내용을 전달하고 있는 부분도 점차 증대되고 있어, 디지털 우편 전자 문서를 장기적으로 보존해야 하는 중요도가 계속 높아지고 있다. 특히, 모바일 기반의 등기 우편 서비스와 같은 법적 효력을 가진 전자 문서 유통도 활성화됨에 따라 단순한 장기보존 이상의 문서 내용의 무결성 및 전송 완료를 보장하는 기술이 계속 발전하고 있다. PDF 형식의 전자 문서는 문서의 무결성을 보장하는 기능 및 디바이스에 독립적인 포맷으로 전자 문서 유통에 널리 사용되고 있으나, 모바일 기반 환경에서는 최적의 사용자 경험을 주지 못한다. 반면, HTML 기반의 전자 문서는 모바일 기반에서 최적의 사용자 경험을 제공하는 형식이기는 하나, 전자 문서라는 측면에서 문서의 무결성 보장이 가능한 장기 보관을 위한 표준은 완벽히 정립되지 않은 상황이다. 본 연구의 목적은 현재 모바일 환경을 위한 HTML 형식의 전자 문서의 중요도가 점점 더 높아지고

Received 26 January 2021; Finally Revised 11 February 2021; Accepted 20 February 2021
† Corresponding Author : wjkim@seoultech.ac.kr

있는 상황에서, HTML 형식의 전자 문서도 효과적으로 생성, 저장 및 무결성 검증이 될 수 있도록, 기존 HTML5 형식을 이용한 Document-HTML 생성 기법에 대하여 설계하고 검증하는 것이다.

본 논문의 구성은 제 1장에서 서론이 기술되어 있으며, 제 2장에서는 본 연구와 관련된 이론적 배경에 대해 기술하였다. 제 3장에서는 Document-HTML 생성 기법의 설계에 대해 기술하였으며, 제 4장에서는 Document-HTML 문서의 검증에 대해 기술하였다. 마지막으로 제 5장에서 본 논문의 결론을 제시하였다.

2. 이론적 배경

2.1 전자 문서

대한민국 전자 문서법에 따르면 전자 문서는 “정보처리시스템에 의하여 전자적 형태로 작성·변환되거나, 송신·수신 또는 저장된 정보를 말한다.”고 정의되어 있다[1]. 일반적으로 전자 거래에서 전자 문서는 전자적 형태의 의사표시를 나타내는 것을 의미하며, 공인전자 문서센터의 전자 문서는 전자적 형태의 서류를 의미한다고 할 수 있다. 이는 디지털 고지서와 같이 기업과 개인 간에 발생하는 전자 문서는 일반적으로 전자적인 형태로 기록될 뿐, 종이 문서와 동일한 형태를 보이는 특성이 있다. 이에 통상 전자 문서란 아래아한글 문서, MS 워드 문서, PDF, HTML, JPEG, TIFF 등 종이 문서와 동일한 형태를 전자적 디스플레이에 표시 또는 종이 문서로 인쇄 가능한 전자 파일 등을 의미한다.

이러한 전자 문서는 <Table 1>과 같이 형식별로 상이한 특성이 있다. 아래아한글과 MS워드와 같은 전자 문서 편집 소프트웨어는 문서 작성에 특화되어 있어서 최초 전자 문서를 작성하는 용이하지만, 여러 종속성 문제를 가지고 있다. 이로 인하여 배포된 문서는 다른 전자 디스플레이에서 다른 형태로 보일 가능성이 높다. 이러한 문제를 해결하기 위해서 아래아한글과 MS워드 등에서 작성된 전자 문서는 외부 종속성이 없고 장치 독립적인 PDF와 같은 형식으로 변환하여 배포하는 것이 일반적이다. PDF 형식은 전자 서명(Digital-Signature)을 파일 안에 포함하여 문서의 무결성을 확인할 수 있는 기능도 제공하고 있다.

CISCO Global Networking Trend 보고서에 의하면 2017년에서 2022년까지의 비즈니스 모바일 트래픽의 성장률이 연 42%로 예측되며, 대다수의 기업이 모바일 기반의 비즈니스로 이동하고 있다[2]. PDF 형식의 전자 문서는 가장 널리 쓰이는 방식이기는 하나, 모바일 환경에서는 작은 스크린 사이즈로 인한 가독성의 어려움과 같은 불편한 사용자 경험 등으로 HTML 문서의 중요성이 점점 더 높아지고 있다.

2.2 종이 우편 서비스 vs. 디지털 우편 서비스

종이 우편 서비스 중 등기 우편 업무는 문서 원본의 발송 확인을 공신력을 가지고 있는 우체국이 인증을 해주는 서비스로서 법적인 지위를 가지고 있으며, 중요한 법적 내용을 포함한 문서들이 등기 우편 서비스로 발송되고 있다. 페이퍼리스 시대가 되면서 종이 우편 서비스에서 제공하고 있는 등기 우편 서비스와 같은 법적 효력을 가진 디지털 문서 유통 서비스도 시작되었으며, 정부 전자 문서 유통 표준이 설립되어 이를 기본 안으로 하고 있다[5]. 이러한 전자 문서 유통 서비스의 대표적인 예로서 카카오페이 고지서 서비스가 있다. 이 서비스는 등기 서비스를 대체하는 법적 지위를 가진 서비스를 제공하고 있다[5]. 이러한 전자 문서 유통 서비스에서는 내용의 무결성 및 전달 완료의 보장을 위하여 인터넷 진흥원과 연계하여 내용의 무결성을 검증할 수 있는 해시값 및 전달 이력을 블록체인에 보관하는 등의 기술로 법적 지위를 보장하고 있다.

<Table 1> Comparison of Characteristics of Digital Document File Format

	HWP	MS Word	PDF	HTML	JPEG	TIFF
Device Dependency	Works on HWP installable OS	Works on MS Word installable OS	No Dependency	No Dependency	No Dependency	No Dependency
Software Dependency	Depend on HWP	Depend on MS Word	Works on Various Softwares	Works on Various Software	Works on Various Softwares	Works on Various Softwares
External Resources Dependency	Need External Resources	Need External Resources	No Dependency	Need External Resources	No Dependency	No Dependency
Editable	Easy	Easy	Difficult	Easy	Difficult	Difficult
Responsive Feature	Nonsupport	Nonsupport	Nonsupport	Support	Nonsupport	Nonsupport
Multi Page	Support	Support	Support	Support	Nonsupport	Support
Vector Graphic	Support	Support	Support	Support	Nonsupport	Nonsupport
Digital Signature	Nonsupport	Nonsupport	Support	Nonsupport	Nonsupport	Nonsupport

디지털 우편 서비스는 이렇게 사용자에게 발송된 전자 문서를 추후에도 원본 그대로 조회 가능하여야 한다는 점이다. 더욱이 법적인 지위를 가지고 있는 문서는 추후 법적 분쟁에서 사용될 수 있어서 문서 생성 시점의 원본 내용을 그대로 재 조회할 수 있어야 한다. 이렇듯 원본 내용의 무결성을 보장하면서 저장 및 조회하는 기능이 더욱 중요해지고 있다.

2.3 PDF

PDF는 종이 문서와 동일한 형태를 디지털 디바이스에 표현할 수 있도록 제정된 표준으로서 다양한 디지털 디바이스에 독립적으로 동일한 형태의 내용을 표시할 수 있도록 한다. 이에 디지털 문서의 신뢰적인 장기 보관을 위해서 추가적인 규약을 더한 PDF/A(PDF for Archive) 표준이 있으며[6, 15], 이는 국가기록원에서 표준 저장 문서 포맷으로 사용되고 있다[12]. PDF/A는 장기 보관 이후에 연결된 리소스의 소실, 비밀번호의 분실 및 열람 소프트웨어의 분실 등으로 해당 문서를 열람하지 못하는 문제를 방지하기 위해 <Figure 1>과 같은 규약을 제시하고 있다. 장기 보관을 위한 PDF/A는 모든 문서를 구성하는 폰트를 포함한 각종 리소스는 반드시 문서 안에 포함되어야 하며, 문서의 암호화(Encryption)는 허용하지 않는다. 그 이외에 종이 문서 역할이 아닌 오디오 및 비디오의 포함 및 자바스크립트 실행(Javascript action) 등은 허용하고 있지 않다. 또한, 전자 서명을 포함한 PDF 문서를 만들 수 있는 규약을 가지고 있으며, 전자 서명이 포함된 PDF 문서는 문서 생성 이후의 위변조 여부를 검증할 수 있다.

- Audio and video content is forbidden
- Dynamic action by Javascript is forbidden
- All fonts must be embedded
- External content reference are forbidden
- Encryption is forbidden
- Device independent colorspace

<Figure 1> Key Elements to PDF/A Conformance

2.4 HTML

HTML은 웹에 콘텐츠를 표현하기 위한 표준 마크업 언어로서 현재 HTML 5.2까지 정의되어 있다[17]. 모바일 환경에서 HTML 문서의 중요성은 계속 커지고 있다. 반응형 웹 기능을 포함하고 있는 HTML 문서는 일반 PC 환경 뿐 아니라 모바일 환경에 적합한 형태로 표현되는 향상된 사용자 경험을 제공하고 있다.

한 화면의 웹 콘텐츠를 표현하기 위해서는 하나 이상의 HTML과 이와 연결된 이미지, 웹 폰트, 스타일 파일 등 여러 리소스 파일들이 연결되어 표현된다. 그렇기 때문에 단순히 HTML 파일만 저장하는 것으로는 장기 보관을 위해 적절하지 않으며, 외부 연결 리소스가 같이 저장되어야 한다. 또한, 외부 리소스와 HTML 파일 간의 동기화가 이루어지어야만 추후 저장된 웹 콘텐츠를 조회하였을 때, 최초 웹에 표현되었던 형태 그대로 볼 수 있다. 이러한 HTML 형식의 특성으로 HTML 전자문서를 단일 파일의 전자 문서로 취급하기에 어려움이 있다.

이러한 어려움은 전자 문서로서의 저장의 관점 뿐 아니라, 일반 HTML 문서의 저장에서 모두 일어나는 어려움이고, 이를 해결하기 위해 웹 아카이브가 연구되고 있다. 웹 아카이브의 목적은 후대를 위해서 현재 웹에서 발생하고 있는 정보를 장기 보관(long-term preservation) 하는 것이다. 웹 아카이브는 고전적인 연구 영역이기는 하나, 후대를 위해 데이터를 보관한다는 관점에서 지속해서 논의되어야 할 중요 영역 중의 하나이다. 웹 아카이브은 (a) 평가(Appraisal) 및 선택(Selection) (b) 취득(Acquisition) (c) 구성(Organized) 및 저장(Storage) (d) 내용 기술(Description) 및 접근(Access)의 4단계로 구성되어 있다[7]. 평가 및 선택 단계에서 아카이브 대상을 선별하고, 취득 단계에서 선별된 대상을 저장하는 과정을 거친다. 이후 구성 및 저장 단계에서 저장된 데이터를 분류 및 정제하고 저장 공간에 보관하고, 다음 내용 기술 및 접근 단계에서 조회를 할 수 있도록 한다. 웹 아카이브의 본 목적이 후대를 위한 장기 보관이었으나, 최근에는 전자문서 사용의 활성화로 법적인 효력을 가진 전자 문서의 전달 이후에 추후 조회를 위한 목적으로도 아카이브 기능이 사용되고 있다. 이를 위해서는 법적인 효력을 가진 전자 문서가 웹을 통해 전달되었을 때, 해당 문서의 원본 진위(Content Integrity)를 보장할 수 있는 아카이브 방법이 고려되어야 한다.

웹에 존재하는 콘텐츠를 취득 단계에서 잘 저장하기 위해서 다양한 웹 저장 표준기법 및 소프트웨어가 존재하고 있다. IIPC는 이러한 웹 컨텐츠를 아카이브하기 위해 ISO 28500 규격의 WARC(Web ARChive) 파일 포맷을 지정하였다[9]. WARC 규약은 웹 사이트 등과 같은 다양한 출처로부터 웹 콘텐츠를 내부 데이터 저장소에 저장 및 관리할 수 있는 방안을 제시하고 있다. WARC 규약은 여러 웹 아카이브를 수행하고 있는 기관에서 채택되어 사용되고 있으며, 한국 국립 도서관에서도 웹 아카이브를 위해 WARC 규약을 사용하고 있다[13]. WARC 규약은 연결된 리소스를 포함하여 저장하는 기능과 본문의 위변조를 검증하기 위한 <Figure 2>와 같이 WARC-Block-Digest 값을

같이 저장할 수 있도록 되어 있다. 하지만, 본문 내용과 해시 다이제스트 값을 동시에 수정하는 경우에는 원본 진위 여부를 확인할 수 있는 방법이 없다. 또한, WARC 규약은 일반 사용자가 사용하는 웹브라우저에서는 확인할 수 없기 때문에 일반 사용자 배포용으로 사용하기에는 어려움이 있다.

MHTML은 RFC 2557 규약에 기반한 MIME HTML을 위한 표준 규격이다[8]. 일반적인 HTML 형식이 HTML 코드로 된 주 파일과 연결된 리소스로 구성되는 반면, MHTML은 모든 연결 파일을 하나의 파일에 포함한다. 하지만, MHTML 규약은 문서의 무결성 검증 방법을 제안하고 있지 않다. 또한 일반 HTML 형식과는 다른 문서 형식으로서, WARC와 마찬가지로 일반 사용자가 사용하기에는 어려움이 있다.

```
<< Definition >>
WARC-Block-Digest = "WARC-Block-Digest" ":" labelled-digest
labelled-digest   = algorithm ":" digest-value
algorithm         = token
digest-value      = token

<< Example SHA-1 labelled Base32 ([RFC3548]) >>
WARC-Block-Digest: sha1:AB2CD3EF4GH5IJ6KL7MN8OPQ
```

<Figure 2> WARC-Block-Digest

2.5 PKI 기반의 전자 서명

PKI(Public Key Infrastructure)는 공개키 암호기술을 통해 사용자 증명을 할 수 있는 비대칭 기반 기술이다[16]. PKI 구조는 공개키와 개인키의 두 부분으로 구성되어 있으며, 개인키를 통해 암호화된 내용은 공개키와 함께 익명의 다수에게 배포가 가능하다. 암호화된 내용은 공개키에 의해 원본 검증이 가능하여, 부인 방지가 이루어져야 하는 전자 계약 및 전자 문서의 무결성 입증에 대중적으로 사용되고 있다. 대한민국에서 널리 사용되고 있는 공인인증서가 PKI 공개키 암호기술 기반으로 구현된 기술이다[14]. PDF 문서 형식에서는 해당 PKI 기반의 전자 서명을 포함하여 해당 문서가 최초 작성된 이후에 위변조가 일어났는지 확인할 수 있는 무결성 검증 기능을 가지고 있으며, DocuSign과 같은 서비스 회사가 PDF 형식 및 PKI 전자 서명 공개키 암호 기술을 이용하여 전자계약 서비스를 제공하고 있다[4].

PKI 공개키 암호기술에서는 암호화를 수행한 개인키가 신뢰할 수 있는 개인키인지 입증하는 것이 중요하다. 공개키를 이용하여 암호화된 내용이 암호화된 이후에 위변조 되지 않은 원본이라고 하더라도, 암호화 시점에 사용된 개인키가 신뢰할 수 없는 개인키라면 암호화된 원본 자체를 신뢰할 수 없게 된다. 이에 모든 공개키, 개인키는 CA(Certificate Authority)라는 인증기관으로부터 발급받아 사용하게 되어 있고, CA 인증기관은 최상위 Root CA 인증기관으로부터 체인(Chain) 형태로 연결되어 신뢰할 수 있는 개인키인지를 증명한다.

3. Document-HTML 생성 기법의 설계

3.1 Document-HTML 선언

HTML 문서가 가지고 있는 모바일 환경을 위한 향상된 사용자 경험을 제공하면서, 신뢰적인 전자 문서로서의 역할을 수행하기 위한 HTML5 기반의 Document-HTML 하위 규약을 정의한다.

해당 HTML 문서가 Document-HTML 규약을 준수하는 문서임을 나타내기 위해서 Document-HTML 확장 태그를 정의한다. Document-HTML 파일 헤더는 문서의 최상단에 위치하여 <html> 태그가 시작되기 전에 나타난다. 기존 HTML 문서와의 호환성을 위해서 해당 확장 태그는 HTML 문서의 주석 안에 정의된다. 각 확장 태그는 <Figure 6>과 같이 <!-BEGIN Document-HTML 확장 태그 시작 문자열과 END Document-HTML-> 종료 문자열 사이에 해당 확장 태그의 값이 들어가도록 한다.

해당 문서가 Document-HTML 문서임을 나타내기 위해서 DOCUMENT-HTML-TYPE 확장 태그를 정의한다. DOCUMENT-HTML-TYPE 확장 태그의 값은 DocHTML5 v0.1 로 본 연구에서는 정의한다.

3.2 Document-HTML을 위한 HTML5 제약 규칙

HTML5 규약은 현재 디지털 채널에서 가장 중요한 콘텐츠 표현 형식이지만, 종이 문서 형태의 문서로서의 보관보다는 디지털 디바이스에서 다양한 정보를 표현하는 것에 더 목적을 두고 있다. 이러한 목적으로 종이 문서를 대체하는 장기 보존 형식으로 사용되기에는 PDF 규약보다는 어려움이 있다. 본 연구에서는 PDF와 같이 장기 보존 및 디지털 채널 특성을 모두 만족하기 위해 Document-HTML을 위한 HTML5 제약 규칙을 지정한다.

(a) HTML 문서의 인코딩은 UTF-8을 규약으로 한다. 장치 독립적으로 문서를 열람하기 위해서 장치 독립적인 문자 인코딩 방식을 UTF-8을 기본 문서 인코딩 규칙으로 한다.

(b) HTML 문서에 필요한 모든 리소스는 문서 안에 포함(self-contained)하며, 외부 리소스를 연결하여 문서를 구성하지 않는다. HTML 문서는 PDF/A와 같은 단일

파일로서 외부 리소스 연결을 허용하지 않는다. HTML 문서를 구성하는데 필요한 CSS, Image, Font, Script 등과 같은 요소들은 모두 HTML 문서 안에 포함되어야 한다.

(b-1) CSS 속성은 HTML 문서 안에 정의되거나 포함되어야 한다. 외부에서 참조된 CSS 속성은 외부 참조 링크 연결 단절 시, 정상적으로 문서를 표현하지 못한다. 이에 CSS 속성은 문서의 HTML 태그 안에 인라인 형식으로 정의되거나, CSS 정의 세트가 문서 안에 포함되어 존재하여야 한다.

(b-2) Image는 모두 HTML 문서 안에 포함되어야 한다. 외부에서 참조된 Image 속성은 외부 참조 링크 연결 단절 시, 정상적으로 문서를 표현하지 못한다. 이에 Image 속성은 문서 안에 포함되어 존재하여야 한다.

(b-3) Font는 웹 폰트(web-font)를 사용하며, 모두 HTML 문서 안에 포함되어야 한다. 일반적인 HTML 문서는 폰트를 포함하고 있지 않다. 문서가 표시되는 장치의 시스템 폰트를 사용하거나, 외부 웹 폰트를 참조하여 표현하고 있다. 시스템에 해당 폰트가 존재하지 않거나, 외부에서 웹 폰트 링크 연결 단절 시, 정상적으로 문서를 표현하지 못한다. 이에 Font 속성은 문서 안에 포함되어 존재하여야 한다.

(b-4) Script는 모두 HTML 문서 안에 정의되거나 포함되어야 한다. 외부에서 참조된 Script 속성은 외부 참조 링크 연결 단절 시, 정상적으로 문서를 표현하지 못한다. 이에 Script 속성은 문서의 HTML 태그 안에 인라인 형식으로 정의되거나, Script 정의 세트가 문서 안에 포함되어 존재하여야 한다.

(c) <audio>, <video> 멀티미디어 요소를 사용하지 않는다. 오디오 및 비디오 요소는 실제 용량의 문제로 인해 하나의 문서로 포함되기 어려운 점이 있으며, 이 요소는 문서로서의 필수 요소는 아니다. 이에 HTML5에서 제공하는 멀티미디어 요소인 <audio>, <video> 멀티미디어 태그는 사용하지 않는다.

(d) <object>, <iframe>, <embed>, <param> 외부 객체 포함 요소를 사용하지 않는다. 외부 해당 리소스는 하나의 문서로 포함되기 어려우며, 장치에 의존해서 기능을 보여줄 수 있고, 이 요소는 문서로서의 필수 요소는 아니다. 이에 HTML5에서 외부 객체 포함 요소인 <object>, <iframe>, <embed>, <param> 태그는 사용하지 않는다.

(e) 비동기 데이터 로딩을 허용하지 않는다. HTML 문서가 실행된 이후, 자바스크립트 등과 같은 스크립트 언어를 이용한 비동기 호출 방식으로 데이터를 적재하는 것을 허용하지 않는다. 이것은 외부 리소스를 불러와서 실시간으로 HTML 문서상에 데이터를 표현하는 것이기 때문에 데이터 자체가 HTML 문서에 포함되어 있지 않다. 이는 HTML 문서의 무결성을 유지하지 못하게 하는 요소가 된다.

3.3 단일 문서로서의 저장

앞에서 정의된 Document-HTML 문서는 모든 외부 리소스가 단일 문서 안에 포함된 리소스로 존재하여야 한다. 문서 안에 인라인 (Inline)으로 정의한 경우가 아닌, 외부에 정의된 CSS, Script와 같은 리소스 또는 이미지 파일 등도 하나의 리소스로 변환되어 포함되어야 한다. 이러한 외부 리소스를 단일 HTML 문서에 포함하기 위해서 <Figure 3>과 같이 HTML5 규약에서 제공하고 있는 Data URL scheme을 사용한다[11].

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

<Figure 3> Data URL Scheme

Data URL은 외부 연결 리소스 대신 HTML 파일 내부에 포함된 리소스를 참조할 수 있는 기능을 제공한다. 연결 리소스에서 사용되고 있는 이미지, 자바스크립트 라이브러리, CSS 파일 모두 mediatype 접두어와 base64 인코딩을 통해 내부 포함된 요소로 변환되어 저장될 수 있으며, 이를 통해 외부 연결 파일로 인한, 무결성 유지 취약점을 해결할 수 있다. 단일 파일로서의 저장 속성을 반영한 Document-HTML 문서의 예는 아래 <Figure 4>와 같다.

```
<!DOCTYPE HTML>
<html>
  <head>
<title>HTML4Doc Sample</title>
<link href="data:text/css;charset=us-ascii;base64,LyohCiAqI.. "/>
</head>
<body>
<div>
  <img src=".. "/>
</div>
</body>
</html>
```

<Figure 4> Example of Document-HTML Structure with Embedded Resources

3.4 Document-HTML 무결성을 위한 확장 태그

Document-HTML은 단일 파일로 구성되어 외부 리소스의 연결 없이 문서의 내용을 원본 그대로 표현할 수 있다. 하지만, 여전히 HTML 문서는 위변조에 취약점을 가지고 있으며, 문서의 신뢰성을 보장하기 어렵다. PDF 형식의 문서에 비해서 HTML 형식의 문서는 악의적인 조작에 노출되기 더 쉽다. HTML 문서는 기본적으로 텍스트 기반 문서로서, 메모장과 같은 텍스트 편집 프로그램에서 손쉽게 파일 수정이 가능하다. 이러한 승인되지 않은 위변조로 인한, 무결성 유지 취약점이 발생한다. 이를 방지하기 위해서는 HTML 문서가 최초 생성 이후에 의도치 않은 위변조를 검증할 수 있는 규약이 필요하다. 이에 문서 신뢰성을 위한 Document-HTML 파일 헤더를 정의한다.

(a) DOCUMENT-HTML-TYPE

HTML 문서가 Document-HTML이라는 것을 나타내기 위한 확장 태그를 정의한다. 이 확장 태그는 Document-HTML 문서 헤더에 기록된다.

(b) DOCUMENT-HTML-CONTENT-DIGEST

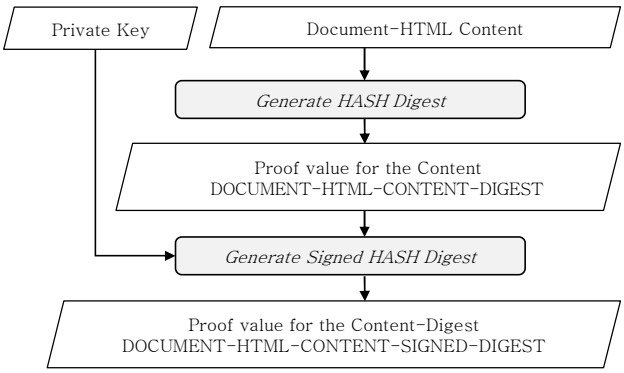
HTML 문서의 무결성 검증을 위한 해시 다이제스트 값을 저장하는 확장 태그를 정의한다. DOCUMENT-HTML-CONTENT-DIGEST 확장 메타 태그는 단일 파일로 변환된 HTML 파일의 해시 다이제스트 값을 생성하여 넣도록 한다. SHA-2 알고리즘은 대표적인 암호화 해시 함수 집합이며, 미국 국가안보국에서 개발된 알고리즘으로 대중적으로 많이 사용되고 있다. 해시 함수 집합의 가장 큰 특징으로는 충돌 회피이며, 모든 파일은 값은 같은 길이의 다른 값을 가진 해시 함수 결과 값을 가진다. 본 연구에서는 SHA256 알고리즘을 사용하여 256bit 길이의 값을 만들어 내고, 이를 텍스트 형태로 변환하여 기록한다. 이 확장 태그는 Document-HTML 문서 헤더에 기록된다.

(c) DOCUMENT-HTML-CONTENT-SIGNED-DIGEST

DOCUMENT-HTML-CONTENT-DIGEST 태그에서 기록된 HTML 문서의 해시 다이제스트 값은 추후 Document-HTML 문서 검증 단계에서 원본 HTML 문서의 원본 진위 확인에 사용될 수 있다. 하지만, 악의적인 사용자에 의하여 원본 HTML 문서의 내용과 해당 DOCUMENT-HTML-CONTENT-DIGEST이 동시에 조작되는 경우에는 이를 확인할 방법이 없다. 이는 여전히 Document-HTML 파일이 원본 검증의 측면에서 취약점을 가지고 있는 것을 알 수 있다. 이를 해결하기 위해서 PKI 기반의 전자서명 기능을 이용하여 해당 DOCUMENT-HTML-CONTENT-DIGEST를 서명한 값을 DOCUMENT-HTML-CONTENT-SIGNED-DIGEST에 기록한다. 이 확장 태그는 Document-HTML 문서 헤더에 기록된다.

(d) DOCUMENT-HTML-CONTENT-VALIDATION-KEY

DOCUMENT-HTML-CONTENT-SIGNED-DIGEST의 값을 검증하기 위한 PKI 기반의 공개키 값을 저장한다. 이 확장 태그는 Document-HTML 문서 헤더에 기록된다.



<Figure 5> Extended Tag Generation Flow

정의된 확장 태그 속성의 생성 과정은 <Figure 5>와 같이 본문 검증용 DOCUMENT-HTML-CONTENT-DIGEST 값 생성 후, 해당 값은 개인키를 이용하여 DOCUMENT-HTML-CONTENT-SIGNED-DIGEST 값을 생성한다. 이렇게 생성된 값은 추후 검증을 위한 공개키와 함께 Document-HTML의 확장 태그로 저장된다. 이를 반영한 Document-HTML 문서의 구성은 아래 <Figure 6>과 같다. Document-HTML의 확장 태그를 문서 최상단에 기록되어 문서의 속성을 나타내고 있으며, 실제 HTML 문서의 시작은 확장 태그 다음 줄 부터이다.

```
<!-BEGIN DOCUMENT-HTML-TYPE
DocHTML5 v0.1
END DOCUMENT-HTML-TYPE->
<!-BEGIN DOCUMENT-HTML-CONTENT-DIGEST
HEXDIGEST_VALUE
END DOCUMENT-HTML-CONTENT-DIGEST->
<!-BEGIN DOCUMENT-HTML-CONTENT-SIGNED-DIGEST
HEXDIGEST_VALUE
END DOCUMENT-HTML-CONTENT-SIGNED-DIGEST->
<!-BEGIN DOCUMENT-HTML-CONTENT-VALIDATION-KEY
HEXDIGEST_VALUE
END DOCUMENT-HTML-CONTENT-VALIDATION-KEY->
<!DOCTYPE HTML>
<html>
  <head>
<title>Document-HTML Sample</title>
</head>
<body></body>
</html>
```

<Figure 6> Document-HTML Structure with Extended Tag

3.5 무결성 검증 구조

Document-HTML 문서는 Document-HTML 확장 태그에 정의된 값을 이용하여 문서의 무결성을 검증할 수 있다. DOCUMENT-HTML-CONTENT-VALIDATION-KEY 값은 PKI 기반의 공개키로써 키 무결성 검증이 가능하며, 이 키를 바탕으로 DOCUMENT-HTML-CONTENT-SIGNED-DIGEST 무결성 검증이 가능하다. Document-HTML 문서는 내용은 DOCUMENT-HTML-CONTENT-DIGEST 값을 통해 무결성 검증이 되며, DOCUMENT-HTML-CONTENT-DIGEST 값은 DOCUMENT-HTML-CONTENT-SIGNED-DIGEST 값을 통해 무결성을 보장받는다. 해당 Document-HTML의 무결성 안전도는 PKI 기반 알고리즘을 사용하기 때문에, PKI 기반 알고리즘 수준의 안전성을 가지고 있다.

4. Document-HTML 문서의 검증

4.1 검증 요소의 선정

해당 Document-HTML 문서가 PDF 형식의 전자 문서와 같이 신뢰적으로 사용될 수 있는지에 검증 요소를 선정한다. <Table 1>에서 비교된 것과 같이 중요 취약점 두 개를 확인할 수 있다. 첫째, 일반 HTML문서는 PDF 문서에 비해 모든 요소가 포함되어 있지 않아서, 모든 리소스를 모두 함께 보관해야만 추후 동일한 문서를 볼 수 있다. 둘째, PDF 문서와 달리 전자 서명에 대한 규약이 존재하지 않아서 문서의 외변조 공격에 대한 취약함을 가지고 있다. Document-HTML 문서가 이 두 가지 취약점을 어떻게 보완하는지를 검증하도록 한다.

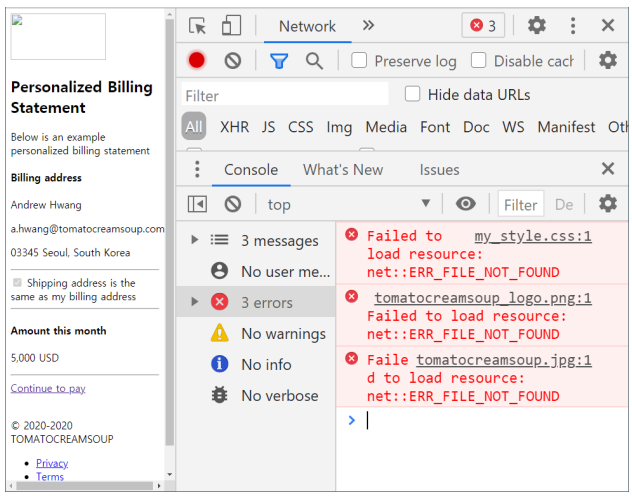
4.2 외부 리소스 연결 없는 독립된 문서의 검증

Document-HTML 전자 문서는 모든 리소스가 포함된 형태로서 외부 리소스 연결을 허용하지 않는다. 이를 검증하기 위해 검증용 HTML 전자 문서를 작성하고 검증하였다. 검증용 HTML에 포함될 HTML 요소를 선정하기 위해서 실제 기업에서 발행되고 있는 HTML 기반의 전자문서를 분석하였다. 대한민국 상위 보험 및 카드사에서 발행되는 전자문서를 분석한 결과 실제 외부 리소스를 참조하기 위해 사용된 HTML 요소는 <img>, <link>, <script>가 사용되고 있었다. 수집된 샘플에서 사용된 요소들을 포함하고 있는 검증용 일반 HTML 전자 문서는 <Table 2>와 같이 웹 폰트 1개, 외부 CSS 파일 1개, 외부 이미지 파일 1개를 포함하고 있는 전자 문서이다. 이렇게 작성된 검증용 일반 HTML 전자 문서를 외부 리소스가 모두 내부 리소스로 변환된 검증용 Document-HTML 전자 문서로 변환한 후, 이 두 문서를 비교 검증하였다. 외부 리소스를 내부 리소스로 변환하기 위해서는 python 언어를 이용하여 변환 스크립트를 개발하여 사용하였다. Document-HTML 전자 문서는 외부 리소스를 모두 문서 안에 포함하고 있어 파일 사이즈가 증가하며, 외부 리소스는 BASE64 인코딩된 문자열로 포함되기 때문에 외부 리소스를 포함한 일반 HTML 파일 사이즈보다 더 많은 용량을 차지한다.

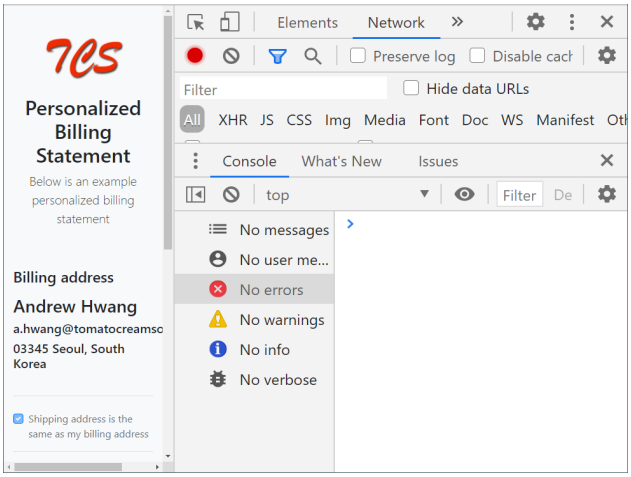
<Table 2> Comparison Normal HTML and Document-HTML

	Normal HTML	Document-HTML
HTML	HTML×1 EA	HTML×1 EA
External Resources	CSS×1 EA Images×2 EA Javascript×1 EA	None
HTML File Size	2,970 byte	1,337,721 byte
Total File Size	1,003,807 byte	1,337,721 byte

이 검증용 일반 HTML과 Document-HTML을 크롬 브라우저의 개발자 도구를 활성화하여 문서의 로딩 상태를 확인한 결과, 일반 HTML은 외부 리소스를 로딩하여 문서를 구성하는 반면에 Document-HTML은 내부에 포함된 리소스로 문서를 구성하는 것을 확인하였다. 외부 리소스 유실의 경우, <Figure 7>과 같이 일반 HTML은 ERR_FILE_NOT_FOUND 오류가 발생하며 정상적으로 표시되지 않는 반면에, Document-HTML은 <Figure 8>과 같이 외부 리소스가 이미 문서에 다 포함되어 있어서 정상적으로 표시되는 것을 확인하였다. HTML5 Data URL규약을 준수하여, 리소스가 HTML 문서 안에 포함된 경우에는, 문서 독립적으로 외부 리소스 요소의 변경 및 유실 등의 영향 없이 항상 동일한 내용을 표현해 주는 것을 확인할 수 있다.



<Figure 7> External Resources Loading Error in Normal HTML



<Figure 8> Embedded Resources Loading in Document-HTML

이미지 파일은 정상적으로 로딩이 되어 이미지가 모두 표시되었으며, 외부 참조 대신, 문서 내에 포함하는 것이 문제가 없는 것을 확인하였다. CSS 파일도 정상적으로 로딩이 되어 각 HTML 요소에 적용된 스타일이 정상적으로 표현되었다. CSS 파일의 경우에도 문서 내에 포함하는 것이 문제가 없는 것을 확인하였다. 자바스크립트 파일도 정상적으로 로딩이 되어 HTML 요소에 적용한 스크립트가 작동하는 것도 확인하였다. 자바스크립트 파일의 경우에도 문서 내에 포함하는 것이 문제가 없는 것을 확인하였다.

4.3 위변조 무결성

Document-HTML 전자 문서는 PKI 기반의 전자 서명을 포함하고 있고, 이를 통해 문서의 원본 무결성을 검증할 수 있다. Document-HTML 전자 문서는 본문 해시 값을 개인키로 암호화한 값과 이를 검증할 수 있는 공개키를 포함하고 있다. 악의적인 위변조를 수행하기 위해서는 Document-HTML 전자 문서의 본문 수정, 본문에 대한 해시 값의 재 생성 및 개인키를 이용한 해시 값의 암호화를 수행하여야만 한다. 하지만 개인키는 Document-HTML 전자 문서에 포함되어 있지 않기 때문에 공개키 및 암호화 값을 이용하여 한 개인키를 역산해 내어야 한다. 전자 서명에 사용되는 RSA 알고리즘은 <Figure 9>와 같이 큰 소수 p, q를 정의한 다음 $N=pq$를 구한다. 이 후 $\phi(N)$을 계산하여 이와 서로소인 정수 e를 찾아 선택한 다음 공개키 $\{N, e\}$을 얻어낸다. $de \equiv 1 \pmod{\phi(N)}$을 통해 d를 구하고, 개인키 $\{N, d\}$를 얻어낸다[10].

사용한 공개키 $\{N, e\}$와 암호문을 알고 있는 상황에서 개인키 $\{N, d\}$를 얻기 위해서는 N을 인수분해 하여 구해야만 한다. 공개키는 $\phi(N)$와 서로소인 e중 하나를 임의로 선택하는 것이고, 개인키는 e를 기반으로 mod 연산을 만족하는 d를 선택하는 것이기 때문에, 이는 일방향 함수이고 역으로 한 번에 계산하는 것은 불가능하고 N 값을 인수분해하여 일일이 계산해 보아야 한다. RSA-140을 인수분해하여 역계산을 하기 위해서는 약 2,000 MIPS-year이 걸리고, 1024bit 길이의 N을 사용하는 RSA인 경우에는 RSA-140보다 49,000,000배 더 어려운 것으로 연구되었다[3]. 본 연구에서 사용된 전자 서명의 경우에는 2048bit 길이의 N을 사용하였고, 현재의 컴퓨터 성능으로는 현실적으로 역계산이 불가능하다. 이에 Document-HTML 전자 문서는 PKI 기반 수준의 문서 무결성 검증을 제공하고 있다.

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

<Figure 9> RSA Key Generation

5. 결 론

PDF 형식의 전자 문서는 현재까지 종이 문서를 대체하기 위해서 가장 많이 사용되는 문서 형식이다. 하지만 모바일 시대에 들어오면서 HTML 형식의 중요성이 더 커지고 있다. 하지만, PDF 형식의 전자 문서와 다르게 현존하는 일반 HTML 기반의 전자 문서는 문서의 무결성 유지 및 장기보관에 부적합한 요소를 가지고 있기 때문에 종이 문서를 대체하기에는 적합하지 않다. 본 연구에서는 기존의 HTML5 규약의 기반으로, 문서의 무결성 유지 및 장기보관이 가능한 문서로서의 특성을 가질 수 있는 Document-HTML 규약을 정의하고 검증하였다.

본 연구를 통해, 관공서 및 기업에서는 모바일 환경에 적합한 사용자 경험 제공, 문서의 무결성 유지 및 장기보관에 가능한 Document-HTML 기반의 전자 문서를 생성할 수 있고, 이를 최종 사용자에게 전달할 수 있을 것으로 예상된다. 또한, Document-HTML를 통해 종이 문서와 같은 법적 지위를 가질 수 있을 것이라고 생각된다. 추후 연구에서는 Document-HTML을 효과적으로 생성 및 검증할 수 있는 Document-HTML Creation & Validation 시스템 구축 방안에 대하여 확장 연구할 수 있도록 한다.

Acknowledgement

This study was supported by the Research Program funded by the SeoulTech(Seoul National University of Science and Technology).

References

[1] Chung, J.M., Key Issues in the Basic Act of Electronic Documents and Electronic Commerce, *Law Review*, 2017, Vol. 27, No. 4, pp. 79-125.

[2] CISCO, Global Network Trends Report, https://www.cisco.com/c/dam/m/en_us/solutions/enterprise-networks/networking-report/files/GLBL-ENG_NB-06_0_NA_RPT_PDF_MOFU-no-NetworkingTrendsReport-NB_rpten018612_5.pdf.

[3] Contini, S., The Factorization of rsa-140, *RSA Laboratories' Bulletin*, 1999, Vol. 10, pp. 1-2.

[4] DocuSign, The eSignature Solution Trusted by Hundreds of Millions of Users, https://www.docusign.com/products/electronic-signature

[5] E-Document Integrated Support Center, Authorized Electronic Document Provider, https://www.npost.kr/pages/intro/intro_0301.jsp.

[6] Evans, T.N. and Moore, R.H., The use of PDF/A in digital archives : a case study from archaeology, *International Journal of Digital Curation*, 2014, Vol. 9, No. 2, pp. 123-138.

[7] Hwang, H.C., Shon, J.G., and Park, J.S., Design of an Enhanced Web Archiving System for Preserving Content Integrity with Blockchain, *Electronics*, 2020, Vol. 9, No. 8, pp. 1-13.

[8] IETF, "RFC 2557, MIME Encapsulation of Aggregate Documents, such as HTML(MHTML)", http://www.ietf.org/rfc/rfc2557.txt

[9] International Internet Preservation Consortium, The WARC Format 1-1, https://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.1.

[10] Lee, S.U., A 2kβ Algorithm for Euler function ϕ(n) Decryption of RSA, *Journal of the Korea Society of Computer and Information*, 2014, Vol. 19, No. 7, pp. 71-76.

[11] Masinter, L., The data URL scheme, https://tools.ietf.org/html/rfc2397.

[12] National Archives of Korea, Confirmed the PDF/A-1 as the Permanent Preservation Format of Government Electronic Records, https://www.archives.go.kr/next/news/pressDetail.do?board_seq=60623.

[13] National Library of Korea, The Website Building Guide for the OASIS Web Archiving System, http://www.oasis.go.kr/about/guide.do.

[14] Park, H.S., Lee, J.H., and Park, S.C., Implementation, Security, and Usability Analysis of Accredited Certificate-based Internet Banking, *Journal of Internet Computing and Services*, 2017, Vol. 18, No. 4, pp. 69-78.

[15] PDF Association, ISO 19005(PDF/A), https://www.pdfa.org/resource/iso-19005-pdfa/.

[16] Son, Y.H., Choi, W.S., Kim, K.H., Choi, H.N., Lee, D.Y., Oh, C.S., and Cho, Y.H., Hybrid PKI Public Certificate Security Method Based on Device ID, *Journal of the Korea Society of Computer and Information*, 2010, Vol. 15, No. 5, pp. 113-124.

[17] W3C, HTML, https://www.w3.org/TR/html/.

ORCID

Hyun Choen Hwang | http://orcid.org/0000-0003-3841-5570

Woo Je Kim | http://orcid.org/0000-0002-1638-645X