사용자 정보를 이용한 Indexed DB 암호화 인증

황우섭*, 박지수**1), 손진곤*
*한국방송통신대학교 대학원 정보과학과
**전주대학교 컴퓨터공학과
wooseob@knou.ac.kr, zetsaver@gmail.com

Indexed DB Encryption Authentication using User Information

Woo Seob Hwang*, Ji Su Park**, Jin Gon Shon*
*Dept. of Computer Science, Graduate School, Korea National Open University
**Dept. of Computer Science and Engineering, Jeonju University

요 약

인터넷의 발전으로 웹 서비스를 사용하는 사용자가 기하급수적으로 늘어났고 사용자에게 다양한 서비스를 제공하기 위해 많은 기술이 등장하고 있다. 서비스를 받는 사용자의 웹브라우저에서도 서버의 많은 기술을 구현할 수 있는 공간을 제공하고 있는데 바로 Web Storage와 Indexed DB이다. Web Storage는 용도에 따라 수 MB 정도를 사용하지만 많은 양의 데이터를 구조화하여 사용한다면 Indexed DB가 적합하다. 하지만 Web Storage뿐만 아니라 Indexed DB 역시 영속적이고 평문의 데이터를 저장하고 있다. 이러한 데이터는 웹 보안에 취약하여 XSS 등의 공격에 사용자의 데이터가 노출되어 탈취되거나 편집되어 악용될 우려가 매우 크다. 본 논문에서는 이와 같은 취약점을 보완하기 위해 운영체제와 디바이스 정보를 이용하여 사용자를 인증하고 암호화하는 기법을 구현하여 성능 평가를 하였다.

Keyword : 사용자, 디바이스 정보, 인증, 암호화, HTML5, Indexed DB, Encryption, Authentication

1. 서론

HTML5는 웹 문서를 기술하기 위한 웹 표준 규약이다. 이 HTML5에는 다양한 기술을 활용하기 위해 여러 종류의 저장 공간을 제공한다[1]. 대표적으로 웹 스토리지에 속하는 로컬 스토리지와 세션 스토리지가 있다[2]. 그러나 저장할 수 있는 용량에 제한점이 있어 많은 양의 데이터를 구조화하는 Index DB를 사용한다[3]. Indexed DB는 평문의 데이터가 영속적으로 저장되어 사용자의 개인 정보 등이 노출될 위험이 있다. 이는 공격자가 XSS 등의 다양한 방법으로 정보를 탈취하거나 편집되어 악용될 수 있다[4]. 이러한 문제를 해결하기 위해 최근 많은 연구가 이루어지고 있는데, 패스워드 기반 암호화 기법이나 핀(비밀번호)을 이용하면서 서버를 통해 인증하는 기법, 혹은 브라우저 확장 기능을 이용하여 프레임워크를 연구하는 기법 등 다양하다[5-10].

최근 연구들에서는 보안 취약점 해결을 위한 암호화를 사용한다. 그러나 이 연구들 또한 사용자 인증을 위해 서버를 이용하거나, 사용자에게 비밀번호를 요청한다. 그러나 보안 강화를 위해 인증 서버를 별도로 구축할 경우 비용적인 문제가 발생하고, 비밀번호를 이용한 인증 절차는 편리성이 떨어지며 key-logger 등의 공격을 고려해야 한다. 따라서 본 논문에서는 이러한 문제들을 해결하기 위해 운영체제와 디바이스 정보를 암호화에 활용하였다.

2. 관련 연구

2.1 Indexed DB

HTML5에서 Indexed DB는 클라이언트에 영속적으로 구조화된 데이터를 저장하며 키와 값으로 관리한다. 또한 데이터는 B-Tree 구조로 되어 있으며 온라인뿐만 아니라 오프라인에서도 사용한다[3]. 이러한 Indexed DB는 다른 Web Storage처럼 평문 형태로 데이터를 저장하여 XSS 등의 공격에 취약하여 데이터가 탈취되거나 변조되어 악용될 우려가 높으므로 보안에 취약한 문제가 있다[4]. 최근 연구에서는 인증 서버를 추가함으로 발생하는 비용의 문제점이나 비밀번호를 이용하는 인증 절차의 편리성 저하와 함께 key-logger 등의 다른 공격에도 고려해야

1) 교신저자

한다[5-10].

2.2 암호화 및 인증

기존 연구에서 사용자 정보를 로컬 스토리지에 이용하여 암호화를 인증한다[11]. 본 논문에서 제안하는 기법은 Indexed DB에서 사용자 정보를 이용하여 기밀성과 편리성을 높인다. 사용자 정보는 운영체제의 현재 사용자와 디바이스 정보를 이용하며, 암호화 인증에 활용할 사용자 정보는 다음과 같다.

- 메인보드(MB)의 일련번호(SN: Serial Number).
- 메인보드(MB)에 존재하는 BIOS(Basic Input Output System)의 일련번호(SN: Serial Number).
- 메인보드(MB) 시스템의 고유식별자 (UUID: Universally Unique IDentifier)
- 운영체제의 현재 로그인된 User ID 정보

위의 사용자 정보 중 사용자가 많이 이용하는 윈도즈 운영체제를 기준으로 registry의 현재 User ID와 디바이스 정보를 사용하여 암호화하고 사용자를 인증한다[12].

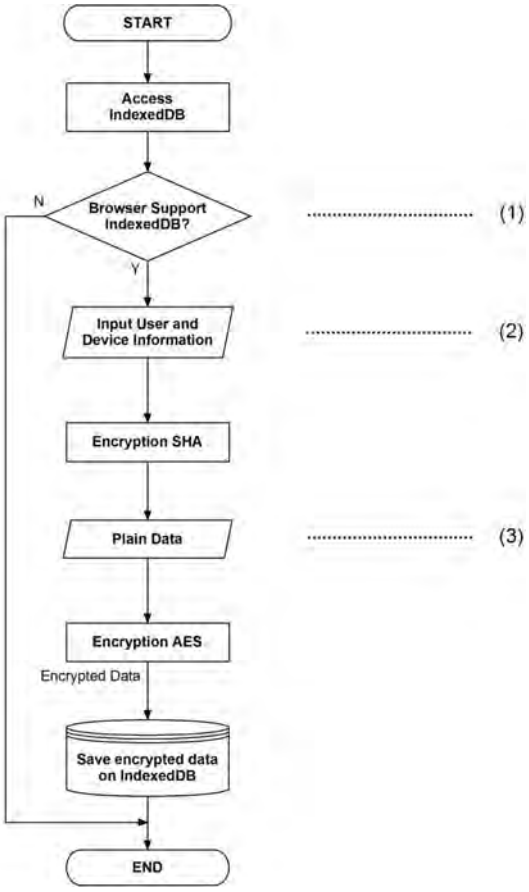
3. Indexed DB 암호화 인증 설계

본 논문에서는 Indexed DB에 저장될 데이터를 암호화할 때 별도의 비밀번호 입력이나 인증할 서버 없이 사용자 정보를 이용하여 암호화하는 기법을 제안한다. 사용자 정보는 운영체제에서 로그인한 현재 사용자 값과 하드웨어의 고유값이며 이 정보를 이용하여 SHA 암호화를 진행한다. SHA로 암호화된 사용자 정보와 평문 데이터를 같이 한 번 더 AES로 암호화한다. 이와 같은 암호화 인증 방식은 데이터가 다른 디바이스로 탈취되거나 변형되어도 안전성이 보장된다. 제안하는 기법의 알고리즘 중 암호화를 설명하면 (그림 1)과 같이 처리되며 설명은 아래와 같다.

(1) 브라우저 응용 애플리케이션이 Indexed DB를 지원하는지 확인한다.

(2) 운영체제의 현재 로그인된 사용자 값을 Registry의 HKey_CURRENT_USER\Identities \User ID에서 가져오고 디바이스 값은 메인보드 내에서 BIOS SN, MB SN, UUID 값을 가져온다. 이 값들은 SHA 256으로 암호화한다. 데이터를 저장할 때 인증할 수 있는 값으로 사용한다.

(3) 데이터를 저장하면 SHA로 암호화된 사용자 값과 평문 데이터 값을 AES 암호화한다.



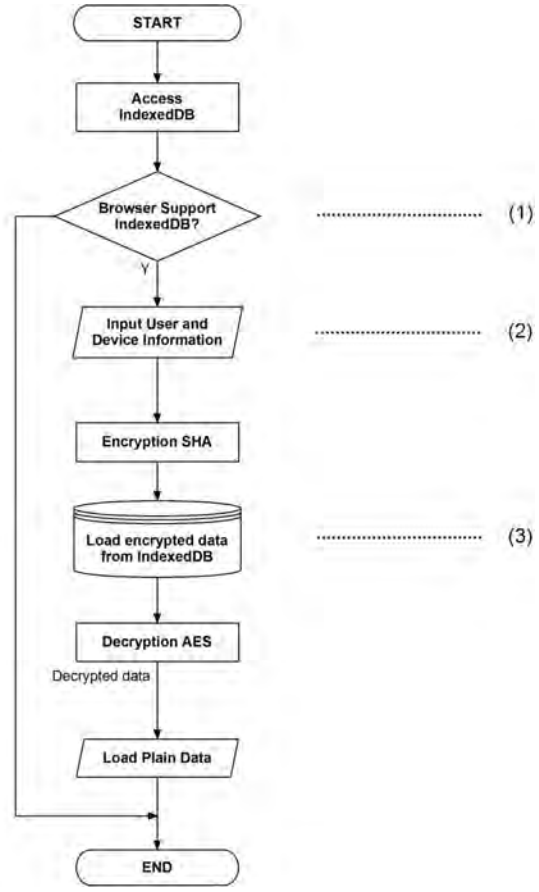
(그림 1) 제안 기법 암호화 흐름도

제안하는 기법의 알고리즘 중 복호화를 설명하면 (그림 2)와 같이 처리되며 설명은 아래와 같다.

(1) 브라우저 응용 애플리케이션이 Indexed DB를 지원하는지 확인한다.

(2) 운영체제의 현재 로그인된 사용자 값을 Registry의 HKey_CURRENT_USER\Identities \User ID에서 가져오고 디바이스 값은 메인보드 내에서 BIOS SN, MB SN, UUID 값을 가져온다. 이 값들은 SHA 256으로 암호화한다. 데이터를 불러올 때 인증할 수 있는 값으로 사용한다.

(3) 데이터를 불러올 경우 Indexed DB에서 암호화된 데이터 값을 얻어오고 SHA로 암호화된 사용자 값으로 AES 복호화 한다.



(그림 2) 제안 기법 복호화 흐름도

4. 성능 평가

본 논문의 제안 기법을 구현하여 테스트한다. 일반적으로 사용자가 많이 접하고 있는 윈도즈 운영체제와 IE(Internet Explorer)로 구현한다. 암호화는 SJCL(Stanford Javascript Crypto Library)를 사용한다[13].

성능 평가는 비밀번호를 기반으로 한 연구와 제안하는 기법과 비교하였다. 그 결과는 <표 1>과 같으며 암·복호화에 걸리는 수행 시간에는 큰 차이를 보이지 않았다. 하지만 사용자의 암호 입력을 기다리는 동안의 시간과 제안 기법의 사용자 정보를 암호화하는 시간에서 암호 입력을 기다리는 시간의 차이만큼 차이를 보여 제안하는 방식이 더 빨랐다.

앞서 언급한 것처럼 윈도즈의 IE 환경에서 검증한 것인 만큼 다른 브라우저에서 구현할 때에는 구현 코드가 달라질 수 있다. Chromium 기반의 Chrome 브라우저에서는 Native Message(메시지교환) 방식이나 NaCl(Native Client)의 PPAPI(Pepper API)를 사용하여 구현해야 한다[14-16].

<표 1> 암호 입력과 제안 기법 비교(단위:ms)

* t : 암호 입력 시간

비교 대상 / 인증 방식	저장 시간 비교			불러오는 시간 비교		
	사용자 인증	암호화	총시간	사용자 인증	암호화	총시간
제안 기법	86.22	5.92	92.14	71.52	5.30	76.82
암호 입력	t	5.17	t+5.17	t	4.69	t+4.69

5. 결론

HTML5의 Indexed DB는 많은 양의 구조화된 데이터들을 활용도 높게 사용할 수 있다. 하지만 평문 형태 데이터를 영속적으로 저장하며 이는 XSS 등의 공격으로부터 목표가 되고 있어 데이터가 탈취되거나 편집되어 악용될 수 있다.

본 논문은 Indexed DB의 기능을 그대로 사용하면서 추가 절차 없이 편리하게 사용하면서 정보를 암호화하여 기밀성을 높였다. 성능 평가도 사용자에게 암호를 요구하여 입력될 때까지 대기시간의 차이만큼 논문에서 제안하는 방법이 빠른 성능을 보였다.

향후 연구에서 최근 많이 발전하고 있는 Web-Assembly를 이용하게 된다면 개발 코드의 큰 변화가 없이 처리 속도까지 향상할 수 있을 것이다[17].

참고문헌

[1] W3C Recommandation, “HTML 5.2”, Available : https://www.w3.org/TR/html52/, Access : april 2020.

[2] W3C Recommandation, “Web Storage (Second Edition)”, Available : https://www.w3.org/TR/webstorage/, Access : April 2020.

[3] W3C Recommandation, “Indexed Database API 2.0”, Available : https://www.w3.org/TR/IndexedDB/, Access : April 2020.

[4] OWASP, “Cross Site Scripting (XSS) | OWASP”, Available : https://owasp.org/www-community/attacks/xss/, Access : April 2020.

[5] J Park, D Shin, D Shin, J Lee, H Lee, “Design and Implementation of Web Browser Secure Storage for Web Standard Authentication

Based on FIDO", Symposium on Information and Communication Technology(SoICT), ACM, Dec 2019.

[6] Amirhossein Akbari, "A NOVEL APPROACH FOR SECURING HTML5 CLIENT-SIDE DATABASE, INDEXEDDB", Tallinn University of Technology, Master thesis, May 2018.

[7] Stefan Kimak, Jeremy Ellman, "The role of HTML5 IndexedDB, the past, present and future", International Conference for Internet Technology and Secured Transactions(ICITST), IEEE, Dec 2015.

[8] Mayssa Jemel, Ahmed Serhrouchni, "Security enhancement of HTML5 Local Data Storage", International Conference and Workshop on the Network of the Future (NOF), IEEE, Dec 2014.

[9] Stefan Kimak, Jeremy Ellman, Christopher Laing, "Some Potential Issues with the Security of HTML5 IndexedDB", International Conference on System Safety and Cyber Security(IET), Oct 2014.

[10] Stefan Kimak, Dr. Jeremy Ellman, Dr. Christopher Laing, "An investigation into possible attacks on HTML5 IndexedDB and their prevention", international Conference on Software, Knowledge, Information Management & Applications(SKIMA), Sep 2012.

[11] 황우섭, 박지수, 손진곤, "사용자 정보를 이용한 인증 절차 자동화", 한국정보처리학회 추계학술대회, 26권 2호, 1125p~1128p, Nov 2019.

[12] Micosoft, "Microsoft Developer Network", Available : https://msdn.microsoft.com/, Search : Registry API, Access : April 2020.

[13] Stanford University Cryptography Group, "Stanford Javascript Crypto Library", Available : https://crypto.stanford.edu/sjcl/, Access : April 2020.

[14] Google Chrome, "Native Messaging", Available : https://developer.chrome.com/extensions/nativeMessaging/, Access : April 2020.

[15] Google Chrome, "Welcome to Native Client", Available : https://developer.chrome.com/native-client/, Access : April 2020.

[16] Google Chrome, "Pepper API Reference (Stable)", Available : https://developer.chrome.com/native-client/pepper_stable/, Access : April 2020.

[17] WebAssembly, "WebAssembly", Available : https://webassembly.org/, Access : April 2020.