한국통신학회논문지 '13-05 Vol.38B No.06
http://dx.doi.org/10.7840/kics.2013.38B.6.473

SHRT : 유사 단어를 활용한 URL 단축 기법

윤 수 진◆, 박 정 은*, 최 창 국**, 김 승 주°

SHRT : New Method of URL Shortening including Relative Word of Target URL.

Soojin Yoon◆, Jeongeun Park*, Changkuk Choi**, Seungjoo Kim°

요 약

단축 URL은 짧은 URL을 긴 URL 대신에 사용하는 방식으로, 짧은 URL이 긴 URL에 리다이렉션되는 방식이다. 단축 URL은 생성과 사용이 간편하고, 글자수가 제한된 마이크로 블로깅 서비스의 사용이 증가함에 따라 폭발적으로 사용량이 증가하였다. 단축 URL의 사용이 간편하기 때문에, 메일, SMS, 책에서도 많이 사용되고 있다. 그러나 대부분의 단축 URL은 연결된 URL과의 어떠한 연관성도 없어, 사용자는 단축 URL에 직접 확인하기 전까지는 무엇에 관한 URL인지 모른다. 연결된 URL을 알 수 없다는 점을 악용하여, 단축 URL은 피싱 사이트나 악성코드 유포 등에 쓰인다. 기존에 이러한 문제를 극복하기 위해 단축 URL 서비스 사이트의 이름을 바꾸거나, 웹사이트의 정보를 반영하거나, 지역 이름의 줄임말 같은 단축어 사용 등의 시도가 있었으나, 각각의 방법에는 자동화의 어려움, 상대적으로 긴 단축 URL 길이, 적용 범위 한계가 각각의 문제점으로 적용하였다. 앞선 문제점을 보완하기 위하여, 본 논문은 아랍어의 모음이 없는 문자 시스템에서 착안하여 URL 사이트 이름에서 모음을 탈락시킨 유사한 문자열을 이용하여 단축 URL 방식 SHRT를 제안한다.

Key Words : URL Shortening, Usable Security, Phishing, Domain, Site Name

ABSTRACT

Shorten URL service is the method of using short URL instead of long URL, it redirect short url to long URL. While the users of microblog increased rapidly, as the creating and usage of shorten URL is convenient, shorten url became common under the limited length of writing on microblog. E-mail, SMS and books use shorten URL well, because of its simplicity. But, there is no relativeness between the most of shorten URLs and their target URLs, user can not expect the target URL. To cover this problem, there is attempts such as changing the shorten URL service name, inserting the information of website into shorten URL, and the usage of shortcode of physical address. However, each ones has the limits, so these are the trouble of automation, relatively long address, and the narrowness of applicable targets. SHRT is complementary to the attempts, as getting the idea from the writing system of Arabic. Though the writing system of Arabic has no vowel alphabet, Arabs have no difficult to understand their writing. This paper proposes SHRT, new method of URL Shortening. SHRT makes user guess the target URL using Relative word of the lowest domain of target URL without vowels.

※ "본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (NIPA-2013-H0301-13-1003)

◆ 주저자 : 고려대학교 정보보호대학원 정보보증 연구실, idtptkd@korea.ac.kr, 학생회원
° 교신저자 : 고려대학교 정보보호대학원 정보보증 연구실, skim71@korea.ac.kr, 종신회원
* 고려대학교 정보보호대학원 정보보증 연구실, planet_in@korea.ac.kr
** 고려대학교 정보보호대학원 정보보증 연구실, necojin@korea.ac.kr

논문번호 : KICS2013-04-186, 접수일자 : 2013년 4월 25일, 최종논문접수일자 : 2013년 5월 8일

Ⅰ. 서 론

SNS(Social Network Service)의 사용이 증가함에 따라, SNS의 한 종류인 마이크로블로그(Microblog)의 사용도 증가하였다. 마이크로블로그는 짧은 글을 간편하게 올려서 사람들 간의 소통을 하는 SNS인데, 대표적인 서비스 사이트로는 트위터(Twitter)와 미투데이가 있다. 마이크로블로그는 쓸 수 있는 글의 길이가 제한되어 있기 때문에 사용자들이 긴 URL을 올릴 때, 표현할 수 있는 글자 수가 줄어들게 되어 많은 불편함이 있었다. 이에 생성과 사용이 간편한 단축 URL(Shorten URL) 서비스가 주목받았다. 단축 URL 서비스는 짧은 URL을 긴 URL로 리다이렉션(Redirection) 시켜주는 서비스로, 마이크로블로그의 발전과 함께 그 사용이 폭발적으로 증가하였다. 단축 URL 서비스가 보편화됨에 따라 간편성을 인정받아 SMS, 메일, 책 등에도 단축 URL 서비스가 많이 생겼다.

그러나 단축 URL 서비스의 사용과 함께 악용 사례도 등장하였다. 대부분의 단축 URL이 클릭하여 리다이렉션 되는 URL을 확인하기 전까지는 사용자가 연결된 URL을 알기 힘들다는 점을 악용하여, 피싱 사이트(Phishing Site)나 유해한 사이트 배포에 쓰이기도 한다. 비록 bit.ly에서 단축 URL 생성 시, 단축 URL의 종류를 분석하여 미리 선점한 단축 URL 서비스 사이트 명으로 교체해주는 방식을 쓰기도 하지만, 모든 URL에 자동화된 방식이 아니라는 단점이 있다. 그 외 통합 선택 URL 단축은 URL내의 정보를 이용하여 단축 URL을 만들지만 어떠한 가공도 없이 정보가 들어가게 되면서 단축 URL의 길이가 상대적으로 길어지게 만든다. 마지막으로 임베디드 단축어를 이용한 시스템과 방법과 지역 주소와 인터넷 주소는 지역의 줄임말을 이용한 단축어를 만들어주지만, 사람들에게 잘 알려지지 않은 경우에는 단축어를 식별하기 어렵다는 문제점이 있다.

앞선 방법들의 단점을 보완하기 위하여, 본 논문에서는 아랍어의 문자 시스템에서 아이디어를 얻은 새로운 단축 URL 생성방법인 SHRT를 제안한다. SHRT는 아랍어의 문자 시스템에 모음이 없고, 글에서 모음을 쓰지 않아도 아랍어 사용자가 모음을 추측해내는 것에 착안하여, 사이트 이름에서 모음을 탈락시킨 유사한 문자열을 단축 URL에 삽입하는 방법이다. SHRT는 이러한 방식을 이용하여 연결되는 URL의 정보를 담으면서도 그 길이를 줄인다.

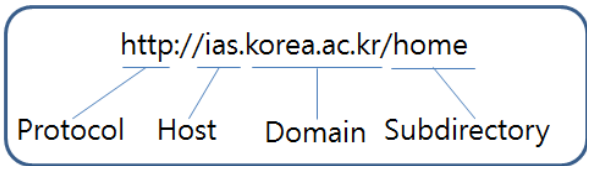

그림 1. URL 구조의 간단한 예
Fig. 1. The simple example of the structure of URL

본 논문은 다음과 같은 구성을 가진다. 제2장에서는 논문에 대한 관련 연구를 설명하고, 제3장에서는 제안하고자 하는 단축 URL 생성법인 SHRT를 설명하고, 제4장에서는 SHRT의 안전성 대해 분석한다. 제5장에서는 실제로 구현된 SHRT를 통해 효율성을 보여준다. 마지막으로 제6장에서는 결론을 짓는다.

Ⅱ. 관련 연구

2.1. URL 구조

URL(Uniform Resource Locator)은 네트워크 자원이 어디에 있는지 알려주기 위한 규약[11]으로 네트워크 자원을 찾아가기 위한 정해진 구조를 가진다. 그림 1은 웹페이지를 가리키는 간단한 URL의 구조이다.

URL은 여러 가지 요소로 이루어질 수 있지만, 많은 경우 프로토콜(Protocol), 호스트(Host), 도메인(Domain), 하위디렉터리(Subdirectory) 등으로 이루어진다.

이 중에서 도메인은 IP 주소를 대신해서 쓰는 문자열로 표현된 인터넷 주소이다. 사이트가 저장된 서버와 같은 네트워크 장치를 찾아갈 때 쓰이는 것이 IP주소인데, IP 주소는 숫자로 이루어져서 사람들이 기억하기 어려워 도메인(도메인 이름)이 등장하였다. 그림 2는 도메인의 계층적인 구조를 보여준다. 가장 위의 단계에는 루트가 있고, 1단계에는 일반최상위도메인과 국가코드최상위도메인이 있다. 2단계 공공도메인을 제외하고 난 후의 최하위 도메인을 사이트 이름이라고 부른다. 예를 들면, 'korea.ac.kr'는 생략된 루트 '.', 최상위도메인은 '.kr', 공공메인인인 '.ac', 최하위 도메인인 'korea'로 이루어져 있다.

국가코드최상위도메인, 일반최상위도메인, 공공도메인은 이미 정해져있다. 도메인 사용자는 정해진 국가코드최상위도메인, 일반최상위도메인, 공공도메인 중에서 선택해야하며, 특별한 목적으로 선점된 경우에는 쓸 수 없다. 사이트 이름의 경우에는 쓰이

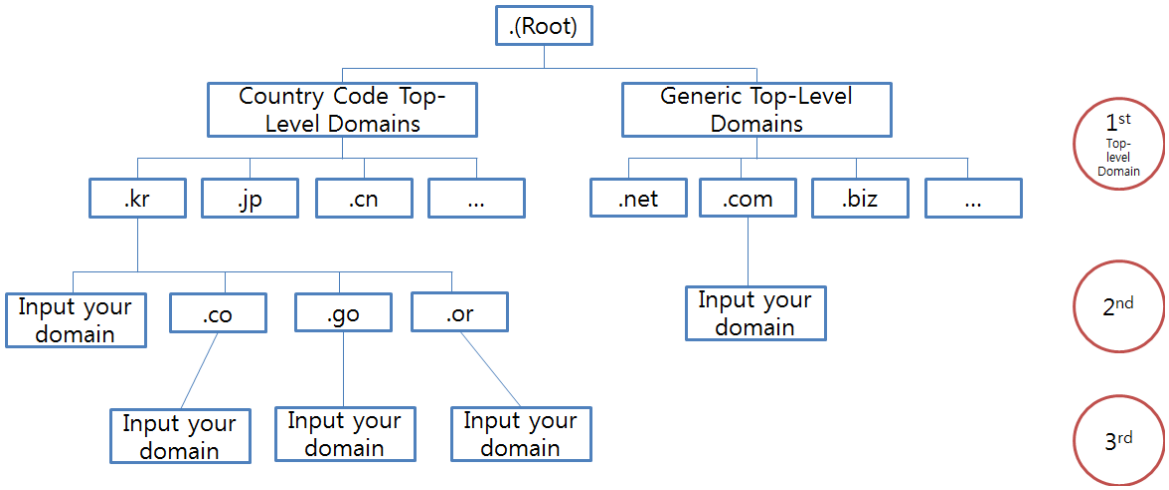

그림 2. 도메인 체계[2]
Fig. 2. The system of domains

는 문자에 따라 다르나, 영문으로 이루어지는 경우에는 영문 대소문자, 숫자, 하이픈(-)으로 구성된다. 단, 하이픈이 첫 글자나 마지막 글자로 올 수 없다. 사이트 이름이 2단계에 있을 때는 3~63글자, 3단계에 있을 때에는 2~63글자까지 가능하다.

2.2. 단축 URL

단축 URL은 긴 URL에 리다이렉션 되는 짧은 URL로 정의되지만, 단순히 짧은 URL을 의미하는 것은 아니다[3]. 연결된 URL을 줄여주는 경우만 단축 URL이라 할 수 있다. 예를 들자면 'https://www.google.com/search?q=ias'는 짧은 URL이지만 단축 URL은 아니고, 'http://urlshorteningservicefortwitter.com/5m6s7'는 단축 URL이지만 짧지 않다.

단축 URL 서비스는 그림 3과 같이 사용자가 단축 URL을 클릭하면 해당 단축 URL 서비스 사이트에서 'HTTP 301 Moved Permanently'와 같은 기능을 이용하여 연결된 URL로 리다이렉션 해주어, 사용자가 단축 URL을 통해서 연결된 URL로 접근할 수 있게 해준다[4,5].

단축 URL의 구조를 살펴보면, 단축 URL 서비스 사이트와 고유 번호로 이루어져 있다. 'shorten.url/PeOH0f'라는 단축 URL를 예로 들면, 'shorten.url'는 단축 URL 서비스 사이트이고, 'PeOH0f'는 단축 URL의 고유 번호이다. 단축 URL 서비스 사이트를 같은 서비스 사이트를 이용하면 같을 수 있지만, 고유 번호는 하나의 단축 URL 서비스 사이트에 내에서 중복되어서는 안 된다. 중복이 될 경우, 한 단축 URL이 여러 URL을 가리킨다는 것인데 이는 보편적인 서비스 제공에 방해가 된다. 만약 'bit.ly/PeOH0f'가 'http://www.kics.or.kr/home/kor/'을 가리키기도 하고, 'https://www.google.co.kr/search?q=KICS'를 가리키기도 한다면, 단축 URL인 'shorten.url/PeOH0f'는 일관성 있는 서비스 제공이 불가능한데, 이는 경우에 따라 가리키는 URL이 다르기 때문이다.

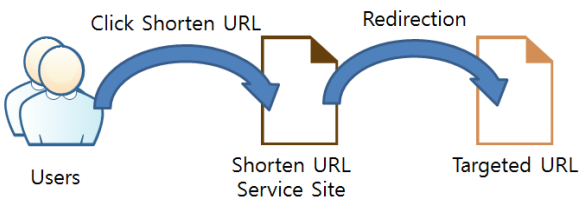

그림 3. 단축 URL 서비스
Fig. 3. Shorten URL Service

단축 URL 생성은 '순서대로 고유 번호를 붙이는 방법'을 주로 쓴다. '순서대로 고유 번호를 붙이는 방법'은 대부분의 단축 URL 서비스 사이트들에서

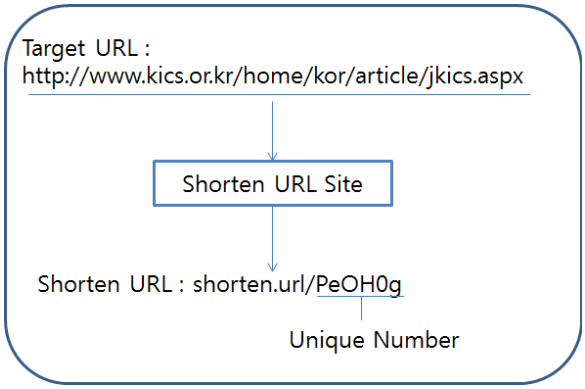

그림 4. '순서대로 고유 번호를 붙이는 방법'의 예
Fig. 4. The example of 'the method of counting unique number as order'

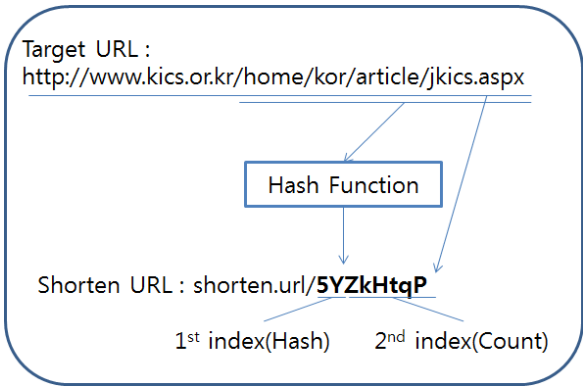

그림 5. '해시 값을 이용하여 번호를 붙이는 방법'의 예
Fig. 5. The example of 'the method of counting unique number using hash value'

제공하는 방법으로 일반적으로 10진법으로 번호를 붙이는 대신 64개의 문자(알파벳 대문자, 알파벳 소문자, 숫자 0~9)를 이용한 64진법을 고유 번호로 쓴다. 단축 URL이 만들어지는 순서대로 번호가 1씩 올라가기에, 'shorten.url/PeOH0f' 후에 만들어지는 단축 URL은 'shorten.url/PeOH0g'가 된다.

이 방법에서 확장된 방법이 '해시(Hash) 값을 이용하여 번호를 붙이는 방법'이다. '해시 값을 이용하여 번호를 붙이는 방법'은 '순서대로 고유 번호를 붙이는 방법'과 유사하나, 단축 URL에 해시 값이 들어가는 것이 특징이다.

해시 함수의 입력 값으로 단축 URL의 하위디렉터리를 쓴다. 해시 값의 길이는 단축 URL 서비스 사이트마다 다를 수 있다. 그림 5는 그러한 경우를 따라 위와 같이 예를 만들었다. 하위디렉터리의 해시 값을 1차 인덱스, 그 뒤에는 연결되는 URL에 대한 2차 인덱스가 단축 URL의 고유 번호가 된다. 해시 값을 이용하여 번호를 붙이는 방식은 많은 양의 단축 URL을 관리할 때, 해시 값을 이용한 검색과 DB 관리를 용이하게 해준다. 만약 단축 URL이 저장된 DB가 여러 개일 경우, 어떤 DB에 접근해야 하는지 결정할 때, 단축 URL에서는 1차 인덱스(원본 URL에서 하위디렉터리의 해시 값)을 통해서 어떠한 DB에 접근할 수 있을지 결정 할 수 있다.

2.3. 단축 URL을 이용한 공격법

단축 URL만을 보고는 연결되는 URL을 알 수 없는 점을 악용하여, 단축 URL을 피싱 사이트나 악성 코드가 포함된 사이트를 유보하는데 쓰이기도 한다. 실제로 단축 URL을 조사한 결과, 피싱 사이트로 연결되는 단축 URL이 있다는 것을 보인 연구도 있다[6,7].

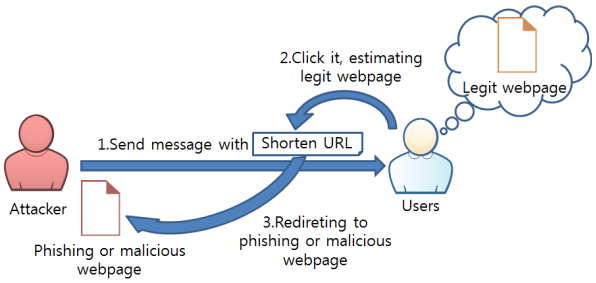

그림 6. 단축 URL을 이용한 공격법
Fig. 6. The attack using shorten URL

그림 6과 같이 공격자는 피싱 사이트나 악성코드가 있는 사이트를 만들어 놓고, 공격자가 만든 사이트와 연결된 단축 URL을 사용자에게 준다. 이 때, 공격자는 사용자에게 공격자의 단축 URL이 사용자가 관심을 가질만한 다른 적법한 사이트의 단축 URL이라고 속이는 메시지를 함께 보낸다. 사용자는 적법한 사이트로 접속을 예상하고 단축 URL을 클릭하게 되고, 공격자가 만든 피싱 사이트나 악성코드가 있는 사이트로 연결이 되어 피해를 입게 된다.

여기서 피싱은 가짜가 진짜인 척 위장하여 개인정보나 금융 정보를 뺏어 악용하는 사회 공학적 기법이고, 악성코드는 사용자의 단말기(PC, 스마트폰 등)에 악영향을 끼칠 수 있는 소프트웨어이다.

단축 URL을 통한 공격에 대해서는 각 단계에서 다음과 같이 방어할 수 있다.

2.3.1. 단축 URL 생성 단계

단축 URL 생성 단계에서 막을 수 있는 방법은 '블랙리스트 확인'과 '사용자가 고유 번호를 지정하여 만드는 방법'이나 '연결되는 URL을 반영한 방법'으로 단축 URL을 생성이 있다.

(1) 블랙리스트 확인

사용자가 원하는 URL을 단축 URL 서비스 사이트에 단축시키고자 할 때, 단축 URL 서비스 사이트에서는 외부나 자체의 블랙리스트를 이용하여 요청 온 URL이 유해한지 혹은 의심스러운 URL인지를 검토한다[8]. 그 후 유해하다고 판단이 되면 단축 URL을 생성하지 않는다. 혹 이미 생성이 되었다가 유해하다고 판단이 되어 사용이 불가능해진 단축 URL이 있다면 이미 차단된 URL임을 알려주며 단축 URL 생성을 막는다.

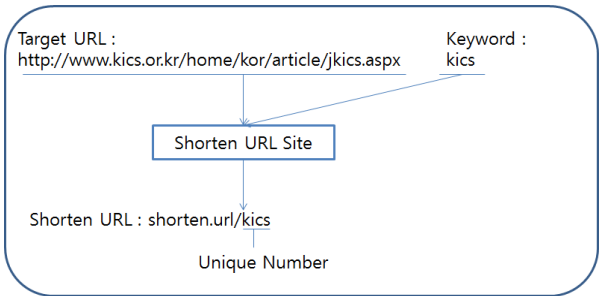

그림 7. '사용자가 고유 번호를 지정하여 만드는 방법'의 예
Fig. 7. The example of 'the method of creating unique as the keyword'

(2) 사용자가 고유 번호를 지정하여 만드는 방법

'사용자가 고유 번호를 지정하여 만드는 방법'은 doiop.com, 3.ly 등 몇몇 단축 URL 서비스 사이트에서만 제공되는 방법으로, 사용자가 입력한 키워드(Keyword)를 단축 URL의 고유 번호로 사용한다. 고유 번호가 선점되어있지 않으면 사용자가 입력한 키워드가 고유 번호가 되고, 선점되어있을 경우에는 불가능하다는 메시지를 보여준다.

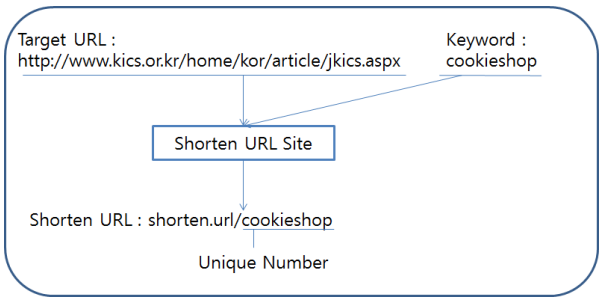

그림 8. 관련 없는 키워드로 단축 URL을 만든 예
Fig. 8. The example of creating shorten URL with unrelated keyword

이 방법은 사용자가 원하는 대로 고유 번호를 설정할 수 있다는 장점이 있지만, 반대로 전혀 연관성 없는 단축 URL을 생성하여 악용될 수도 있다.

(3) 연결되는 URL을 반영한 방법

'연결되는 URL을 반영한 방법'은 연결되는 URL의 정보를 단축 URL에 삽입하는 방법으로, 단축 URL을 통해 연결되는 URL을 추측할 수 있다는 것이 장점이다. 이 방법에 해당하는 방법으로는 'bit.ly의 서비스', '특허 통합 선택 URL 단축[9]', '임베디드 단축어를 이용한 시스템과 방법과 지역 주소와 인터넷 주소[10]'가 있다.

첫째로 'bit.ly의 서비스'로 단축 URL 서비스 사이트를 다르게 표현하는 방법이다. 연결하는 URL

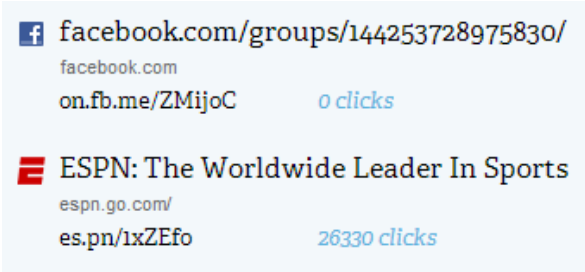

그림 9. 페이스북과 ESPN을 bit.ly에서 단축한 예
Fig. 9. The example of shortening URLs from facebook and ESPN

의 도메인을 살펴서 단축 URL 서비스 사이트를 다르게 표현해주는 방법이다.

페이스북(Facebook), ESPN, 미국 정부 기관 사이트(.gov를 최상위 도메인으로 쓰는 사이트) 등은 bit.ly의 단축 URL에 서비스 사이트를 다르게 표현한다. 페이스북은 'on.fb.me'로, ESPN은 'es.pn'으로, 미국 정부 기관 사이트는 '1.usa.gove'로 표현한다.

둘째로, '통합 선택 URL 단축(Integrated Adaptive URL-Shortening Functionality)'은 연결되는 URL의 정보를 반영시킨 단축 URL 후보를 만들어서 택하게 하는 특허이다. 통합 선택 URL 단축에서 단축 URL에 반영하는 정보는 도메인, URL 경로(Path), 해당 웹페이지의 텍스트 정보 등 이다.

예를 들어 'http://www.bignewspaper.com/breaking-news/mainview/headlines.html'에 대해 ①'short.com/123', ②'short.com/bignewspaper.com/85', ③'short.com/bignewspaer.com/breaking-news' 같은 단축 URL 후보군을 제시한다.

어떤 URL이든 관련 정보가 단축 URL에 있기 때문에 단축 URL와 연결되는 URL 사이의 유사성이 매우 높은 방법이다.

마지막으로 '임베디드 단축어를 이용한 시스템과 방법과 지역 주소와 인터넷 주소(Systems and Methods for Creating and using Imbedded Shortcodes and Shortened Physical and Internet Addresses)'는 키워드에서 두 글자 정도를 뽑아 단축어(Shortcode)로 만들어서 DB에 저장하여 연결해주는 특허이다. 예를 들어 'China.Beijing.Dongcheng.mapic.com'은 'CN.BJ.DNG.mapic.com'으로 줄여준다.

이 방법은 사람이 자주 쓰는 단축어를 이용한다는 점에서 효율적이지만, 유명한 지역 명칭과 같이 단축어가 사회적으로 통용되어야 하는 문제가 있다.

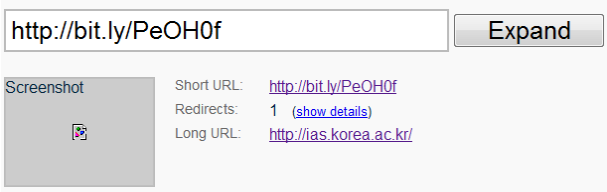

그림 10. longurl.org에서 단축 URL을 확장한 예
Fig. 10. The example of extending shorten URL by longurl.org

2.3.2. 사용자가 단축 URL을 클릭 전

사용자가 단축 URL을 접했을 때, 세 가지 방법으로 사용자가 단축 URL을 확인해보는 방법이 있다.

첫째는 단축 URL을 확장시켜주는 사이트를 이용하여 들어가기 전에 확인해보는 것이다. 이 경우 단축 URL을 확장하여 예상했던 URL과 일치하는지 확인 할 수 있다. 단, 사이트에 들어가서 사용자가 확인해야한다는 번거로움이 있다.

두 번째는 브라우저의 확장기능(Plug-in)을 사용하는 방법이다. 대표적인 확장 기능인 파이어폭스의 bit.ly preview(https://addons.mozilla.org/en-US/firefox/addon/bitly-preview)는 클릭을 하기 전에 커서를 단축 URL에 올려두는 것만으로 연결되는 URL을 알 수 있다.

마지막으로 연결되는 URL을 반영한 방법으로 생성된 단축 URL의 경우를 들 수 있다. 연결되는 URL을 반영한 방법으로 생성된 단축 URL은 사용자가 연결되는 URL을 추측하여, 피싱 사이트나 유해한 사이트에 들어가지 않게 정보를 제공한다.

2.3.3. 단축 URL 클릭 후 리다이렉션 단계

사용자가 단축 URL을 클릭하여 단축 URL 서비스 사이트에서 리다이렉션을 요청할 때 막는 방법이다. goo.gl의 경우에는 단축 URL 생성 단계에서는 막지 않으나, 단축 URL 클릭 후 리다이렉션 단계에서 블랙리스트를 검토하여 리다이렉션을 막는다.

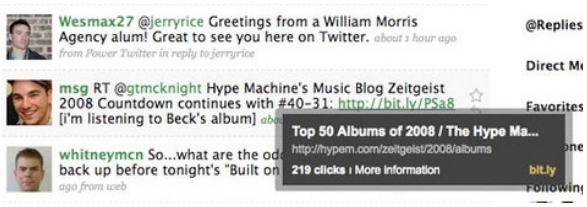

그림 11. 파이어폭스의 확장기능 bit.ly preview 예
Fig. 11. The example of Firefox Plug-in bit.ly preview

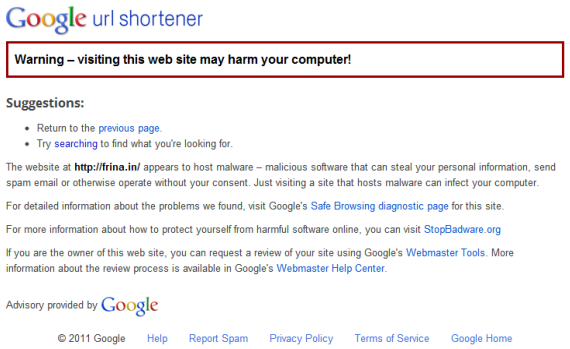

그림 12. goo.gl에서 유해한 사이트에 연결 시도 결과
Fig. 12. The result of trying to connect malicious site through goo.gl

그림 12과 같이 유해한 사이트로 연결되는 단축 URL을 클릭하면 유해한 사이트로 연결을 막는다는 경고 메시지를 띄우기도 하고, 아예 해당되는 단축 URL을 삭제하여 서비스를 불가능하게도 만든다. 국내에서 사용자 학습과 인증기관을 통한 검증이 제안된 바[11]는 있으나 실제로 사용되지는 않았다.

Ⅲ. SHRT

SHRT는 연결되는 URL과 단축 URL과의 연관성이 없는 문제점을 해결하기 위한 새로운 단축 URL 생성 방법으로, 아랍어의 문자 시스템에서 모음이 없는데도 사용자들이 모음을 유추한다는 점에서 착안하여, 사이트 이름의 모음을 탈락시킨 유사한 문자열을 고유 번호에 삽입하는 방법이다. 도메인은 웹페이지 콘텐츠와의 관련성이 매우 높기 때문에[12], URL에서 사람들이 가장 주목하기 쉬운 최하위 도메인(사이트 이름)을 단축 URL에 반영하여, 단축 URL의 문제점을 해결하고 기존의 방법들의 한계점을 보완한다.

3.1. 생성 방법

먼저 줄이고자 하는 URL에서 최하위 도메인을 추출한다. 최하위 도메인에서 자음을 기존에 있던 순서를 맞춰서 뽑아 도메인과 유사한 문자열을 만든다. 이는 모음을 탈락시키는 경우와 같다. 예를 들어, 사이트 이름이 'korea'라면, 'kr'이 유사한 문자열이 된다. 만약 최하위 도메인이 자음으로만 된 경우에는 숫자나 모음을 뽑는 식으로 추가적인 방법을 적용한다. 이렇게 만든 유사한 문자열을 유사 단어(Relative word)라 부르자. 이 유사 단어를 단축 URL의 하위디렉터리를 만들거나 유사 단어를

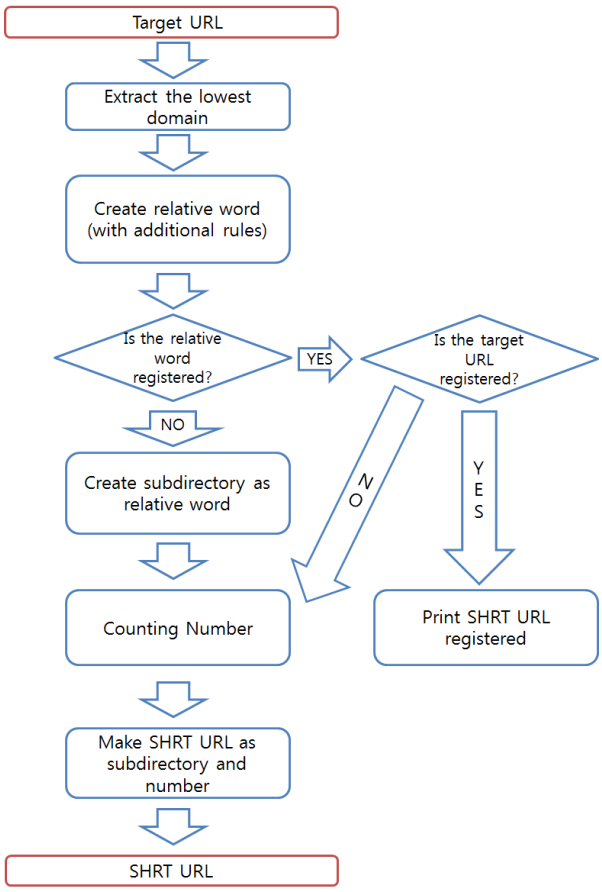

그림 13. SHRT URL 생성 알고리즘
Fig. 13. The algorithm of creating SHRT URL

구분할 수 있는 방식을 취한다. 만약 유사 단어가 등록되어있으면 하위디렉터리를 만드는 작업은 생략한다. 그 뒤 줄이고자 하는 URL이 이미 신청되었는지를 확인 후에 신청되었을 경우, 연결된 단축 URL을 주고, 신청되어있지 않을 경우에는 유사 단어의 뒤에 해시 함수를 이용하여 번호를 붙이는 방법을 사용하여 번호를 매긴다.

3.2. SHRT URL 구조

그림 14는 SHRT 방식을 통해 만들어지는 URL

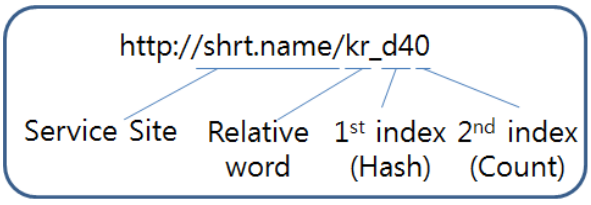

그림 14. SHRT URL 구조
Fig. 14. The structure of SHRT URL

의 구조로, 유사 단어를 하위디렉터리로 만드는 대신에 구분할 수 있는 문자를 써서 표현하였는데, 여기서는 언더바(Underbar)를 사용하여 구분하였다.

SHRT URL은 단축 URL 구조와 같이 서비스 사이트와 고유 번호로 이루어진다. SHRT URL의 고유 번호는 유사 단어, 유사 단어와 인덱스를 구분하는 기호, 인덱스로 이루어진다. 인덱스는 1차 인덱스와 2차 인덱스로 이루어지는데, 1차 인덱스는 연결된 URL의 하위디렉터리 해시 값이고, s2차 인덱스는 유사 단어와 해시 값이 같은 기등록된 URL 개수이다. 인덱스는 알파벳 대소문자와 숫자를 써서 64진법으로 표현된다.

SHRT URL의 고유 번호 구조는 그림15와 같다.

그림 15의 단위는 바이트이다. 유사 단어의 최대치는 사이트 이름의 최대치와 같으므로 63 바이트로 잡았다. 그 뒤 언더바는 구현에 따라 달라 질 수 있으나 유사 단어와 인덱스를 구분하는 기호면 되고, 이 기호에는 1 바이트를 할당했다. 이어서 인덱스는 DB의 크기와 유사 단어 간의 충돌을 감안하여 길이를 가정하였다. 1차 인덱스는 단축 URL이 저장된 DB를 찾는데 쓰이므로, DB의 크기를 감안하여 2 바이트, 2글자가 할당되었다. 2차 인덱스는 유사 단어 간의 충돌을 감안하여 1 바이트, 1글자가 할당되었다. 이후 선택 필드는 나중을 대비하여 만들었다. 실제 구현에서 할당하기 좋기 위하여 8바이트(64비트) 배수에 맞추었다.

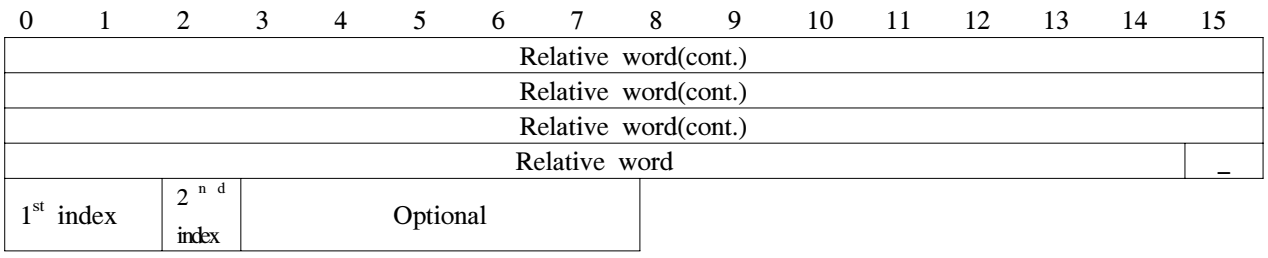

그림 15. SHRT 고유 번호 구조
Fig. 15. The structure of SHRT unique number

3.2.1. DB를 감안한 1차 인덱스 길이

1차 인덱스는 '해시 값을 이용하여 번호를 붙이는 방법'에서 쓰인다. 단축 URL을 저장한 DB가 여러 개일 때, 1차 인덱스 값에 따라 단축 URL이 저장된 DB를 찾을 수 있다. DB의 크기는 1차 인덱스 값 하나에 할당되는 URL의 개수를 통해 가늠할 수 있다. 기존 단축 URL 서비스에 연결된 URL의 개수를 추정하여 DB의 크기를 가늠해보자.

단축 URL에 연결된 웹페이지의 개수를 구하기 위해 14개의 단축 URL 서비스를 통해 살펴본 결과, 단축 URL에서 서비스 사이트 이름을 제외한 번호의 길이는 4~14글자이고, 평균은 6.2글자다. 평균 6.2글자를 기준으로 단축 URL에 연결된 웹페이지의 개수를 구할 수 있는데, 한 글자당 가능한 문자는 62개이므로, 등록된 단축 URL은 $62^{6.2} \approx 1.3 \times 10^{11}$개이다.

DB의 크기에 대한 적절한 기준을 찾기 위하여, 상용 DB의 크기를 살펴보도록 하자. Microsoft사에서 제공하는 Exchange Server의 2010년 일반판(Standard edition) 기준으로 DB의 최대 크기의 기본 설정을 1TB로 하였다[13]. SHRT 구현을 위해 사용한 단축 URL 오픈 소스인 Yourls은 하나의 단축 URL이 처음 만들어질 때, 메타 데이터까지 합치면 2KB 정도의 데이터를 생성한다. 1TB는 2KB로 표현되는 데이터를 5×10^8개 표현할 수 있는 값이다.

만약 해시 값이 1바이트로 하여 1개의 글자로만 표현하게 하면, 한 DB가 감당해야 할 양은 $62^{5.2} \times 2\mathrm{KB} \approx 4.2\mathrm{TB}$이다. 이는 권고된 DB 사이즈보다 매우 큰 값이다. 해시 값을 2바이트로 하여 2개의 글자로 표현하게 되면, 한 DB가 저장해야 할 양은 $62^{4.2} \times 2\mathrm{KB} \approx 67\mathrm{GB}$로 줄어든다. 67GB는 이후에 다른 메타데이터 값들이 생기더라도 1TB안에서 해결될 수 있는 크기이다.

그러므로 해시 값은 2바이트가 되어야 하며, 1차 인덱스의 길이는 2글자가 된다.

3.2.2. 충돌을 감안한 2차 인덱스 길이

2013년 4월 8일을 기준으로 등록된 도메인의 수는 144,818,617개이다[14]. 임의의 웹사이트로 연결해주는 'randomwebsite.com'을 이용하여 100개의 사이트를 수집한 결과, 이 중에서 blogspot.com이 두 번 나타났으나 정작 유사 단어가 중복되는 경우는 존재하지 않았다. 구한 샘플의 비율이 편향적이므로 plus-4 method를 이용하여 중복되는 경우에 +2를, 중복되지 않는 경우에 +2를 하여, 중복되는 확률 $\hat{\pi} = \frac{2}{104}$를 구한다. 이를 통해 등록된 도메인 중에서 유사 단어가 중복된 도메인은 $144,818,617 \times \hat{\pi} \approx 24,784,973$개가 된다. 유사 단어가 중복된 도메인을 제외하여 유사 단어가 중복되지 않는 도메인의 개수를 구하면 142,033,644개이다.

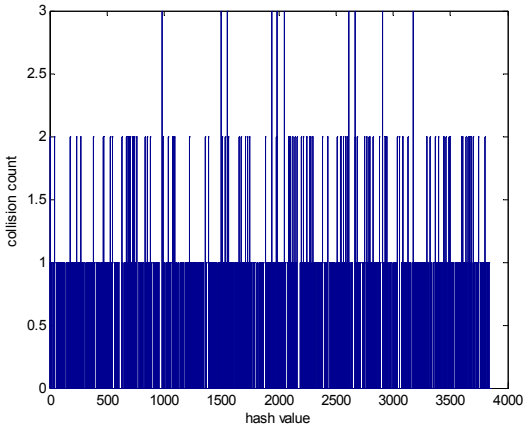

그림 16. 해시 값을 913번 실험한 충돌 횟수 히스토그램
Fig. 16. The histogram of collision count of hash value after 913 iterations

단축 URL들이 모든 유사 단어에 균일 분포(Uniform distribution)로 나누어져있다고 가정하면, 하나의 유사 단어에 등록된 URL의 개수는 약 913개가 된다. 비록 해시 함수는 균일 분포를 가지고 있으나, 이는 실제 환경에서는 충돌이 일어난다.

그림 16은 1차 인덱스를 2byte, 2글자로 표현했을 때 가능한 64^2가짓수를 균일 분포로 913번 실험한 값이다. 실험 결과, 해시 값 사이에 충돌이 있을 수 있으므로 2차 인덱스가 필요하다는 것을 알 수 있다. 위의 결과에 따라 2차 인덱스는 한 글자를 표현할 수 있는 1 byte로 설정하였다.

Ⅳ. SHRT 안전성

단축 URL을 이용한 공격법에서 공격자가 단축 URL을 이용하여 피싱 사이트나 유해한 사이트로 유도하는 경우, 사용자는 예상하고 있는 적법한 사이트의 도메인을 알고 있지만, 확인 할 수 없기에 단축 URL에 접속하게 된다. 그러므로 단축 URL을 사용하는 공격자는 어떠한 도메인이든 피싱 사이트

나 유해한 사이트로 만들 수 있다. 실제로 대검찰청 홈페이지 ‘www.spo.go.kr’에 대한 피싱사이트 중에 전혀 관련 없어 보이는 ‘www.yhofho.com’도 있다. 이만큼 도메인이 완전히 달라도 피싱 사이트로 쓰인다.

그러나 SHRT는 도메인의 유사 단어를 사용하기 때문에, 유사 단어가 중복되는 경우가 아니라면 단축 URL을 이용한 공격을 할 수 없다. 만약 ‘www.korea.ac.kr’를 공격하려면 유사단어인 ‘kr’을 가지는 또 다른 도메인을 준비해야 한다.

그러므로 SHRT가 안전하려면 같은 유사 단어를 가지는 경우가 적어야 한다. 그러나 SHRT는 최하위 도메인의 자음을 추출하여 유사 단어를 만드는 방법으로, 다른 최하위 도메인이더라도 같은 유사 단어를 가질 수 있다.

SHRT가 안전하려면 이렇게 유사 단어가 중복되는 경우가 매우 적어야 한다. 이를 위해 가능한 경우의 수를 비교한 결과와 100개의 도메인을 임의로 추출하여 실험한 결과를 통해 유사 단어의 중복이 어려운 일이며, SHRT가 안전함을 살펴보자.

4.1. 가능한 경우의 수를 통해 본 유사 단어 중복 확률

어떠한 도메인이 있을 때, 그 도메인과 같은 유사 단어를 가질 확률을 구해보자.

특정 유사 단어를 가질 수 있는 도메인의 개수는 유사 단어의 길이에 의존한다. 왜냐하면 SHRT는 자음을 추출하므로 제거되는 16가지(모음, 숫자, 하이픈)를 최대 (63 - (유사 단어의 길이))만큼 섞을 수 있기 때문이다. 유사 단어의 길이를 n라고 하면, 길이가 n인 특정 유사 단어를 가질 수 있는 경우의 수는 수식 (1)과 같다.

$$\sum_{i=0}^{63-n} 16^i \times {}_{i+n}C_n \tag{1}$$

수식 (1)을 통해 얻은 경우의 수를 전체 가능한 도메인의 수로 나누면 유사 단어 중복 확률을 얻을 수 있다. 가능한 도메인의 수는 37가지(알파벳, 숫자, 하이픈)로 표현되는 3글자에서 63글자로 만들 수 있는 문자열의 가짓수로, 이는 수식 (2)와 같다.

$$\sum_{i=3}^{63} 37^i \approx 6.4 \times 10^{98} \tag{2}$$

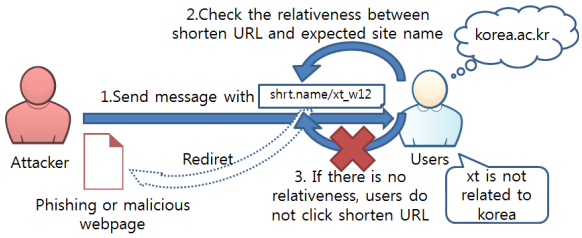

그림 17. SHRT를 이용한 피싱 예방 예
Fig. 17. The example of preventing from phishing with SHRT

유사 단어가 중복될 확률은 수식 (1)을 통해 구한 가짓수를 수식 (2)의 값으로 나누면 얻을 수 있다. 표 1는 그러한 값을 나타내었다.

4.2. 100개의 도메인을 통해 본 충돌 확률

위에서 살펴본 대로 산술적으로 유사 단어의 충돌이 매우 힘들다는 것을 알 수 있다. 실제에서도 그러한지 알아보기 위한 실험은 다음과 같다. 임의의 사이트로 연결해주는 ‘randomwebsite.com’을 이용하여 100개의 사이트를 수집하여, 유사 단어를 만들어 본다. 이후 만들어진 유사 단어가 일치하는 쌍의 개수를 센다.

자음을 추출한 결과, 같은 유사 단어를 가지는 쌍은 한 쌍밖에 나타나지 않았다. 단순히 보자면 1%의 충돌 가능성을 가지고 있다. 그러나 이도 ‘blogspot.kr’에 속하는 두 개의 블로그가 나와서 생긴 경우이다. 즉 실제 충돌로 보기 힘들다. 그러므로 SHRT에서 유사 단어가 충돌할 확률은 매우 낮아 안전하다.

추가적으로 SHRT는 자음을 추출하였는데, 반대로 모음을 추출한 경우를 같은 샘플에 적용시켜보자, 100개 중에서 총 40개가 충돌을 일으켰고, 심지어 3개가 같은 하나의 유사 단어를 가지는 경우도 ‘ae’, ‘e’, ‘ea’, ‘oe’, ‘oeea’로 다섯 가지나 되었다. 즉, SHRT에서 모음을 추출한 방법을 쓴다면 유사 단어의 충돌이 너무 잦아서 안전하지 않다.

도메인의 유사 단어가 중복될 확률이 매우 낮으므로, SHRT는 단축 URL을 이용한 공격법을 막을 수 있다. 단축 URL을 이용한 공격이 가능한 이유는 연결되는 URL과 단축 URL 사이의 연관성이 없다는 점인데, 이러한 문제는 순서대로 번호를 붙이는 방식과 해시 값을 이용하여 번호를 붙이는 방식에서 발생한다.

단축 URL을 이용한 공격법의 경우, 공격자가 피싱 사이트나 악성 코드가 있는 사이트의 웹페이지

표 1. 유사 단어의 길이에 따른 유사 단어 중복 확률
Table 1. The probability of collision or relative word according to the length of relative word

the length of relative word	the probability of collision of relative word	the length of relative word	the probability of collision of relative word	the length of relative word	the probability of collision of relative word
1	4.74×10^{-23}	22	2.01×10^{-33}	43	2.60×10^{-59}
2	9.18×10^{-23}	23	2.24×10^{-34}	44	7.38×10^{-61}
3	1.17×10^{-22}	24	2.33×10^{-35}	45	1.94×10^{-62}
4	1.09×10^{-22}	25	2.27×10^{-36}	46	4.75×10^{-64}
5	8.04×10^{-23}	26	2.07×10^{-37}	47	1.07×10^{-65}
6	4.85×10^{-23}	27	1.77×10^{-38}	48	2.23×10^{-67}
7	2.47×10^{-23}	28	1.42×10^{-39}	49	4.27×10^{-69}
8	1.08×10^{-23}	29	1.07×10^{-40}	50	7.46×10^{-71}
9	4.11×10^{-24}	30	7.57×10^{-42}	51	1.19×10^{-72}
10	1.39×10^{-24}	31	5.03×10^{-43}	52	1.71×10^{-74}
11	4.17×10^{-25}	32	3.14×10^{-44}	53	2.22×10^{-76}
12	1.13×10^{-25}	33	1.84×10^{-45}	54	2.56×10^{-78}
13	2.77×10^{-26}	34	1.02×10^{-46}	55	2.62×10^{-80}
14	6.17×10^{-27}	35	5.25×10^{-48}	56	2.34×10^{-82}
15	1.26×10^{-27}	36	2.55×10^{-49}	57	1.79×10^{-84}
16	2.36×10^{-28}	37	1.16×10^{-50}	58	1.16×10^{-86}
17	4.07×10^{-29}	38	4.96×10^{-52}	59	6.12×10^{-89}
18	6.49×10^{-30}	39	1.99×10^{-53}	60	2.55×10^{-91}
19	9.59×10^{-31}	40	7.44×10^{-55}	61	7.83×10^{-94}
20	1.32×10^{-31}	41	2.61×10^{-56}	62	1.58×10^{-96}
21	1.68×10^{-32}	42	8.52×10^{-58}		

URL을 단축 URL로 만든 후 사용자에게 보낸다. 기존의 단축 URL은 사용자가 단축 URL을 부가적인 방법(브라우저의 확장기능, 단축 URL 확장 사이트)을 이용하여 확인하거나 '연결되는 URL을 반영한 방법'으로 단축 URL을 생성하지 않고는 단축 URL이 어디에 리다이렉션 되는 지 알 수 없다. 그러나 SHRT의 경우에는 사용자가 단축 URL과 자신이 예상하는 사이트의 최하위 도메인 주소를 비교할 수 있다. 사용자가 SHRT URL과 자신이 예상하는 URL을 비교 후에 유사성이 있다고 생각할 경우에는 단축 URL을 클릭하여 넘어가지만, 유사성이 없을 경우에는 클릭을 하지 않아, 악성코드에 감염되어 피해를 입거나, 피싱 사이트를 통하여 금전적 피해를 입을 수도 있는 상황을 피할 수 있다.

Ⅴ. SHRT의 효율성

SHRT의 효율성은 실제 단축 URL 서비스 사이트와 길이를 비교하면 알 수 있다. 단축 URL 오픈소스인 Yourls를 이용하여 SHRT를 구현(http://shrt.name)하였고, 'randomwebsite.com'를 이용하여 수집한 100개의 사이트의 구현된 SHRT URL의 고유번호의 평균 길이를 알아보았다. 그림18은 SHRT를 구현한 shrt.name를 이용하여 'http://www.kics.or.kr/home/kor/article/jkics.aspx'를 단축한 예이다.

shrt.name에 100개의 URL을 넣어 실험한 결과, 사이트 이름을 제외한 번호의 길이는 평균 10.0개의 글자로 나타내는 것을 보였다.

다른 방법에서 고유 번호의 길이를 살펴보면, '순서대로 고유 번호를 붙이는 방법'의 경우에는 평균 6.2글자, '통합 선택 URL 단축'의 경우에는 도메인을 추출하였을 때는 평균 13.3글자, '임베디드 단축

그림 18. shrt.name의 사용 예
Fig. 18. The snapshot of shrt.name

어를 이용한 시스템과 방법과 지역 주소와 인터넷 주소' 경우에는 최소 2글자가 소요된다. 비록 SHRT의 고유 번호의 길이는 10.0글자이지만, 만약 사이트의 이름을 알고 있다면, 유사 단어와 언더바를 제외하면 3글자만 알고 있어도 SHRT URL을 기억할 수 있다.

Ⅵ. 결 론

본 논문에서는 기존의 단축 URL이 연결되는 URL과 연관성이 없다는 점에서 발생하는 문제점에 주목하였다. 단축 URL과 연결되는 URL 사이에 연관성이 없기에 사용자는 단축 URL을 들어가기 전까지 본인이 예상하는 URL에 연결되었는지 알 수 없다. 이런 문제점을 보완하려는 방법들이 있으나, 각각 문제점을 안고 있다. SHRT는 이러한 문제점들을 보완하는 새로운 단축 URL 생성법으로, 연결되는 URL의 최하위 도메인에서 자음만을 추출하여 유사 단어를 만들어서 단축 URL에 넣는 방법이다. SHRT는 기존의 단축 URL의 기능을 제대로 수행하면서도, 단축 URL의 보안 메커니즘 상 보완하기 어려웠던 사용자가 단축 URL 클릭 전 단계에서 사용자가 쉽게 단축 URL을 확인할 수 있게 하였다.

그러나 자음만 추출된 유사 단어와 연결된 URL을 추측하지 못 하는 상황이 발생할 수 있다. 'korea'가 주어졌을 때, 'kr'과의 유사성을 찾기는 쉽지만, 'kr'만 주어졌을 때 'korea'를 떠올리기는 힘들 수 있다. 그러나 기존의 공격 방법에서는 속이는 메시지가 있기 때문에 사용자는 비교할 수 있는 대상이 있어서 추측하기 어렵지 않다. 또한 아예 다른 사이트로의 접속을 유도하였기 때문에 SHRT는 이러한 공격을 막을 수 있다.

References

[1] IETF, *RFC 1738 Uniform Resource Locators(URL)(1994)*, Retrieved Apr., 8, 2013, from http://www.ietf.org/rfc/rfc1738.txt

[2] KISA, *Domain system(2013)*, Retrieved Apr., 8, 2013, from http://domain.kisa.or.kr/jsp/domain/domainInfo/domainSys.jsp

[3] A. Neumann, J. Barnickel, and U. Meyer, "Security and privacy implications of URL shortening services," in *Proc. Workshop on WEB 2.0 SECURITY AND PRIVACY (W2SP) 2011*, pp. 1-10, Oakland, U.S.A., May 2011.

[4] N. Megiddo, P. Alto, and K. S. McCurley, "*Efficient Retrieval of Uniform Resource Locators*," U.S. Patent No. 6,957,224 B1, Oct. 2005.

[5] D. Antoniades, I. Polakis, G. Kontaxis, E. Athanasopoulos, S. Ioannidis, E. P. Markatos, and T. Karagiannis, "We.b: the web of short urls," in *Proc. 20th Int. Conf. World Wide Web*, pp.715-724, Hyderabad, India, Mar. 2011.

[6] D. K. McGrath and M. Gupta, "Behind phishing: an examination of Phisher modi operandi," in *Proc. 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET'08)*, article no. 4, San Francisco, U.S.A., Apr. 2008.

[7] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: the underground on 140 characters or less," in *Proc. 17th ACM Conf. Comput. Commun. Security (CCS'10)*, pp.27 - 37, Chicago, U.S.A., Oct. 2010.

[8] G. Kontaxis, I. Polakis, M. Polychronakis, and E. P. Markatos, "dead.drop: URL-Based Stealthy Messaging," in *Proc. 7th European Conf. Comput. Network Defense (EC2ND)*, pp.17-24, Gothenburg, Sweden, Sep. 2011.

[9] E. Vishria, S. Carlos, T. Howes, L. Altos, and R. Churehill, "*Integrated Adaptive URL-Shortening Functionality*," U.S. Patent No. US 2011/0264992 A1, Oct. 2011.

[10] S. L. Hancock, "Systems and methods for

creating and using imbedded shortcodes and shortened physical and internet addresses," U.S. Patent 2011/0244882 A1, Oct. 2011.

[11] J. Park, S. Yoon, and S. Kim, "A Phishing link prevention for QR code and shortend URL on smart phone," in *Proc. CISC-S'12*, pp. 241-245, Cheonan, Korea, June 2012.

[12] I. Jo and H. Y. Yeom, "Evaluation of the domain-content relevancy as a key feature of Phishing detection," in *Proc. Conf. KICS*, pp. 789-791, Suwon, Korea, Nov. 2011.

[13] Microsoft, *Microsoft Exchange Server - Modify a Database Size Limit(2010)*, Retrieved Apr., 8, 2013, from http://technet.microsoft.com/en-us/library/bb232092.aspx

[14] Whois Source, *Domain Counts & Internet Statistics(2013)*, Retrieved Apr., 8, 2013, from http://www.whois.sc/internet-statistics

윤 수 진 (Soojin Yoon)

2012년 2월 고려대학교 정보통신컴퓨터공학부 졸업

2012년 3월~현재 고려대학교 정보보호대학원 석사과정

<관심분야> SNS 보안, Usable Security

박 정 은 (Jeongeun Park)

2011년 2월 대구가톨릭대학교 컴퓨터공학과 졸업

2011년 8월~현재 고려대학교 정보보호대학원 석사과정

<관심분야> SNS 보안, 웹 보안, 정보보증

최 창 국 (Changkuk Choi)

2001년 2월 광운대학교 화학공학과 졸업

2000년~2002년 마크로테크놀러지 대리

2002년~2005년 시큐아이 대리

2010년~현재 NHN 차장

2012년 2월~현재 고려대학교 정보보호대학원 석·박사 통합과정

<관심분야> 해킹, CCTV 보안

김 승 주 (Seungjoo Kim)

1994년 2월 성균관대학교 정보공학과 졸업

1996년 2월 성균관대학교 정보공학과 석사

1999년 2월 성균관대학교 정보공학과 박사

1998년 12월~2004년 2월 KISA(舊한국정보보호진흥원) 팀장

2004년 3월~2011년 2월 성균관대학교 정보통신공학부 조교수, 부교수

2011년 3월~현재 고려대학교 사이버국방학과.정보보호대학원 정교수

2012년 3월~2012년 6월 선관위 디도스 특별검사팀 자문위원

<관심분야> 보안공학, 암호이론, 정보보증, 정보보호제품 보안성 평가, Usable Security