"Social TAN" - A Privacy-Enabling One-Time Short URL Service

Ulrich König

Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein (ULD), Germany
ULD61@datenschutzzentrum.de

Abstract: The social TAN service provides a one-time URL shortening service. It enables the user to keep in touch with people she meets, using her social network profile with the security that the shared information will only be usable for a single session for one user. At the end of the session, the link to the information is destroyed.

1 Introduction

Privacy-aware people nowadays do not want to be findable by everybody in the Internet. On the other hand, they want to be able to stay in touch with people they know. They establish new connections to people they get to know in real life or virtual places.

In the past, many people used their full real name to find each other in social networks. Today the majority of privacy-aware people just use their first name or a pseudonym. The problem is that people they meet in real life are unable to find each other in a social network if both persons are privacy-aware.

One solution could be that both persons exchange the pseudonyms they are using. Nevertheless, these pseudonyms are not always communicated clear without ambiguities.

Another way would be to exchange the direct link to the social network profile. However, this is not useful, because these Uniform Resource Locator(s) (URL) use to be very long. This leads to typing errors. In addition a pseudonym, direct link or e-mail address can be transferred to a third person who should not get this piece of information.

The social TAN service provides a solution for this problem. It enables the user to provide a short URL that links to one of her identities in a social network. The URL is limited to just one user in just one session. A person who has received such a URL can use it once to get access to the linked social network profile. After having used that URL, it does only provide access to the profile within the same session for a limited time. The social TAN links will destroy itself after usage.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4

2 Basic functionality

The social TAN service is based on two techniques. Both will be described in this section.

2.1 TANs and other one-time (or few-times) credentials

The classic transaction authentication number (TAN) is known from its usage in banking environment. The basic idea is that the user gets a list of numbers. Each one of these numbers is only used one-time, to authenticate. This procedure prevents replay attacks because each number can only be used once. So, a used TAN is worthless.

2.2 URL shortening service

Nowadays many website URLs are machine generated. The objective of this machine generated URLs is often to be search engine friendly, semantically human understandable or just easy to implement. Implementing URLs as short as possible is rarely done, even if long URLs usually are inconvenient, hard to remember or write down. They often cause problems in chats, web forums, e-mail or twitter messages.

Often URL shortening services are used to compress long URLs. The user enters a long URL and gets in return a short substitute URL. The new URL points to the URL shortening service. When it is accessed, the URL shortening service looks up the related URL and redirects the user to the destination of the long URL.

3 Related work

The social TAN service is based on a one-time password system defined in [Hal94] or [LMS05].

The second part of the social TAN service is the URL shortening service that is described in [APK+11].

To identify unique users, techniques introduced in [Eck10] are used.

The capacity of the human working memory was presented in [Mil56] with 7 ± 2 chunks.

4 Architecture

This section will explain the architecture of the social TAN service. It will address what terms are used in which meaning and how the social TAN link is generated. The details how the social TAN service works and the structure of the underlying database will be explained.

4.1 Terms and definitions

The used terms will be defined in this section.

4.1.1 Publishing User

The publishing user is a user who generates a social TAN link from an original URL with help of the social TAN service and transfers it to another user, the viewing user.

The URL could be a link to the personal social network profile.

4.1.2 Viewing User

The viewing user is a user who gets a social TAN link from the publishing user to can get access to the link the publishing user originally has put into the social TAN service.

4.1.3 Primary Key

The primary key is a unique identifier for every social TAN link within the social TAN service. It is built from letters (a-z,A-Z) and numbers (2-9). The look-alike characters (0,O,1,I,l) will be excluded, so that 57 characters will be usable. Optionally, special characters such as (!, @, #, $, %, &, *, (,), -, +, =, ., :, „ ;, ”, ’, <, >, [,], {, }) could be used.

Equation 1 shows how to the possible combinations can be calculated. The example uses 5 symbols with a hamming distance of 16 bit with $a :=$ number of symbols in alphabet, $l :=$ length of primary key and $d :=$ hamming distance. The example shows that 5 symbols contain enough information to encode 268 million different social TAN links with an average hamming distance of 16 bit (1:65536). The amount of symbols can be increased at any time, if the system runs out of combinations.

$$2^{\lceil \log_2 a^l \rceil - \frac{d}{8}} = 2^{\lceil \log_2 57^5 \rceil - \frac{16}{8}} = 268,435,456 \quad (1)$$

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4

4.1.4 Social TAN Link

A social TAN link is a URL pointing to the server from the social TAN service. The URL contains a fix prefix for example `http://socialtan.net/` and the primary key. The social TAN link should be transferred from the publishing user to the viewing user. A complete social TAN link may look like this: `http://socialtan.net/IDdqd`

4.1.5 Social TAN Management Link

The social TAN management link is a social TAN link with a longer primary key. It is used for authentication and enables the publishing user to control the social TAN links that she has generated before. This is similar to the 4chan tripcodes as described in [BMHH+11].

The authenticated publishing user can see which social TAN links have been used. All of her unused social TAN links can be revoked or the destination be changed.

It is also possible to provide a general management interface with username and password login, to manage all generated social TAN links or use the social network login via single sign-on with OpenID.

4.1.6 UUID

The objective of the unique user identification (UUID) (or universally unique identifier [LMS05]) is to identify user A as exactly as possible to prevent other users to impersonate user A. On the other hand, the UUID can also be used to identify attackers. To identify attackers' characteristics only features of the attackers may be used, that cannot be modified easily by the attackers themselves.

For example, it is not easy to change the IP address more than a few times. Nevertheless, IP addresses are often used by more than one person. In case of radio access network (RAN) like Universal Mobile Telecommunications System (UMTS), many users use just one IP address. So only blacklisting by IP address may lock a high number of users out from using the service. This may change with IPv6. [Eck10], [1]

4.1.7 Session

A session is the time a viewing user is identified by the same UUID. A session lasts 24 hours at the maximum. The objective is to limit one session to one person at one location.

A simple example, would be to generate an UUID from the IP-Address and a unique cookie in the users browser.

[1] Panopticlick: `https://panopticlick.eff.org`

ISBN 978-3-88579-286-4

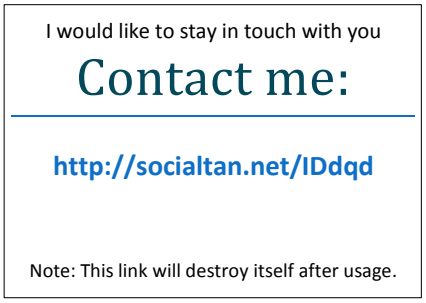

Figure 1: Example of a business card to deploy a social TAN link.

4.2 Generating social TAN links

The publishing user generates a list of social TAN links in three steps:

1. Access the social TAN service root page. E.g.
   `http://socialtan.net`
2. Fill in URL to social network profile and click on the "Generate" button.
3. Store output of social TAN link. The publishing user gets:
   - a list of the social TAN links and
   - one special social TAN management link to manage the generated social TAN links (revoke, update destination, see status).
   - the option to print the social TAN links on business cards, with her own printer. See Figure 1 for an example.

4.3 Deployment of social TAN links

Although a social TAN link consists of just a few symbols, it is on the border of the human working memory [Mil56], depending on the person. More than one social TAN link is too much for the human working memory, if the symbols are not pooled into chunks.

To not push the memory of the users more than needed, it is recommended to write the social TAN link down on a sheet of paper and carry it for example in the wallet or the business card case. For users convenience the social TAN service provides an easy way to printout the generated social TAN links. An Example can be found in Figure 1.

4.3.1 Limitations

There are some limitations concerning generation of social TAN links. These should prevent one or a small number of publishing users generate many social TAN links so the social TAN service runs out of primary keys. For more information about misuse of the social TAN service, see section 5 Attacks.

- One URL (prefix) can be assigned up to 100 active social TAN links.
- One UUID can generate up to 1000 social TAN links per 24 hours.

4.3.2 social TAN link encoding

The social TAN link will contain the primary key as postfix.

All primary keys will have a hamming distance [Ham50] of 16 bit or more in average. The primary keys may not have a constant hamming distance, because then it would be much easier for a malicious viewing user to guess primary keys an invalidate social TAN links randomly.

4.3.3 Long term usage

In long term perspective, the social TAN service will run out of combinations for primary keys. So the social TAN service has to increase the length of the primary key. However this also decreases the usability for the users, because they have to write down or enter longer URLs.

A simple way to limit the number of used primary keys is to define a time to life for every social TAN link. A good value seems to be one year. Nevertheless, there are no empiric values to verify that one year is a good choice, so it is more a random chosen value. However, it is important, if a social TAN link has a limited time to life, to inform the publishing user for how long the social TAN link is going to be valid.

4.4 Internal processing and security of a Lookup

If the social TAN link is accessed by the viewing user, the social TAN service will look up the primary key in the database. If it is valid, the viewing user will be redirected to the URL the publishing user has deposited as shown in figure 2.

4.4.1 Algorithm

The social TAN service will distinguish between two Cases: case A: correct redirection of user and case B: error handling. Which case will occur depends on four questions.

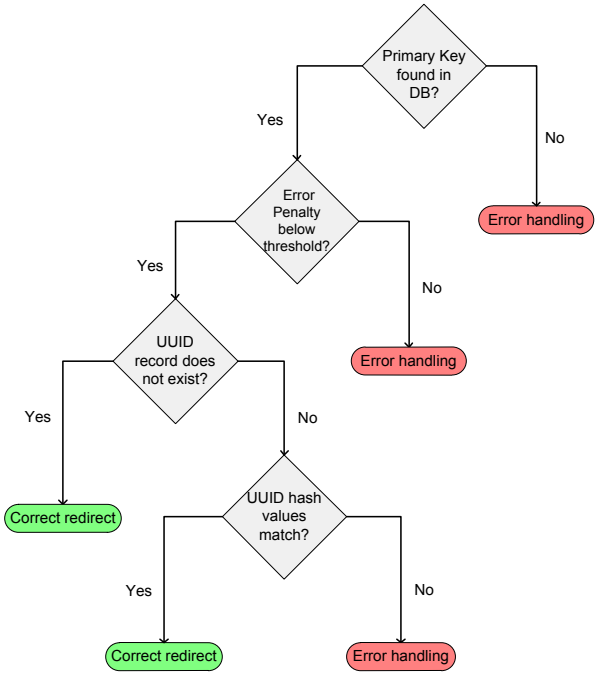

Figure 2: The social TAN service program flow.

1. Is the primary key found in Database (DB)?
   - If no: case B: error handling will occur.
   - If yes: go on to 2.
2. Is the error penalty below threshold?
   - If no: case B: error handling will occur.
   - If yes: go on to 3.
3. UUID record does not exist?
   - If no: go on to 3.
   - If yes: case A: correct redirection of user will occur.
4. UUID hash values match?
   - If no: case B: error handling will occur.
   - If yes: case A: correct redirection of user will occur.

4.4.2 Case A: Correct redirection of user

The viewing user will be redirected to the URL, stored in the database.

4.4.3 Case B: Error handling

The viewing user will get an error message.

In addition, the error penalty for the UUID of the viewing user will be increased by one.

4.4.4 Error penalty

The error penalty is a metric to test if a viewing user is trying to manipulate the service or if she is just using it in the way, it is meant to be.

Every time the error penalty is increased by one, the following data will be stored:

1. UUID
2. count
3. TimeToLive

The whole error penalty record for the UUID will be deleted if the error penalty for the UUID has not been increased for 24 hours.

By default, the error penalty for unknown viewing users is zero and ok. If the error penalty is three or higher, the error penalty is not ok.

If the error penalty is not ok, the viewing users will always get an error message as described in case B: error handling.

4.5 Database architecture of the social TAN service

The database schema will contain two tables and four columns. Both tables will be linked by the primary key field.

4.5.1 Social TAN table:

- primary key

  The primary key field is the primary key for the database table and the social TAN service.

- URL

  The URL field contains the target URL for the social TAN link.

- UUID

  The UUID field contains at creation time `NULL`. When the social TAN link is accessed the first time, the UUID of the viewing user will be saved in this field.

- TimeToLive

  The TimeToLive field contains the time, when the record will be deleted. This time will be updated to $now() + 24h$, when the social TAN link will be accessed the first time, by the viewing user.

4.5.2 Social management table:

- long primary key

  The long primary key identifies the social TAN management link.

- primary key

  The primary key identifies the social TAN link, which is managed by the social TAN management link.

A search for a specific social TAN management link primary key returns all associated primary key for social TAN links.

4.5.3 Error penalty table:

- UUID

  The UUID field contains the UUID of the viewing user that accessed the social TAN service with an invalid primary key.

- Count

  The count field contains the quantity how often the viewing user has accessed the social TAN service with an invalid primary key.

- TimeToLive

  The TimeToLive defines the timestamp, when the viewing user will be rehabilitated and her error penalty record be deleted.

4.6 Implementation effort

For testing reasons, a prove of concept implementation has been created and is available for testing at [2]. This "prove of concept" implements only the very basic subset of functions that are described in this paper. The implementation took a few hours and is about 100 lines of code long. For a productive scenario it would take more time to implement and test the security mechanisms, but it should be doable within a few person months.

[2]Social TAN - prove of concept prototype: `http://www.socialtan.net/`

4.6.1 Integration into running Social Network

To implement it into a social network, it is only necessary to change the prefix from `http://socialtan.net/` to `http://exampleSNS.com/t/`. The `t/` is the internal prefix for social TAN links into the social network. For this kind of integration, an unused internal (the part of the prefix behind the domain name) prefix is needed. A complete social TAN link would look like this: `http://exampleSNS.com/t/IDdqd`.

It is also possible to use basic authentication, as described in [FHBH+99], to transfer the social TAN link. A basic authentication social TAN link could look like this: `http://IDdqd@exampleSNS.com`. The difference is that the primary key in this example moves in front of the domain name. This will work, if the social network is not using basic authentication or the uses usernames are collision free with the social TAN links. Anyway, the basic authentication link has the problem, that users may think the social TAN link is an e-mail address, because there is a @ sign in the given URL.

4.6.2 Working expenses

The costs for server are neglectable, since the social TAN serviceuses a very simple linear logic with no loops. The most computing intensive operation is the lookup in the Database for UUID and Primary Keys, which can be handled very fast by a NoSQL as described in [LM10], since there are no joins included.

5 Attacks

This section deals with possible attacks. It will address the prevention or confinement of consequences.

5.1 To illegally access the service

5.1.1 Brute-Force Attack

The attacker accesses social TAN links with random primary keys or all primary keys one after each other.

To prevent this, the primary keys has a hamming distance of 16 bit. This results in a probability of $< \frac{1}{65536}$ to hit an existing primary key. Every time the attacker hits a not existing primary key, the error penalty will be increased. If the error penalty is three or higher, the attacker will get the same reaction as if the accessed primary key does not exist. So the probability to access one valid primary key with brute-force and one UUID is $< \frac{3}{65536} = \frac{1}{21845,333}$ per 24 hours.

5.1.2 Distributed brute force attack

When the attacker is able to switch it UUID on every time the social TAN service locks one UUID, the social TAN service is defenceless. This can happen if an attacker is using a botnet or a cloud-service. Nevertheless, an attacker can only access the social TAN service three times with a wrong primary key per UUID. The attacker would still need a minimum average of 21845 fresh UUIDs per malicious accessed social TAN link, if the maximum social TAN links are stored in the social TAN service. So the expenses for an attacker to compromise one social TAN link are very high. This may protect the social TAN service.

5.2 To disable the service

5.2.1 (Distributed) Denial of Service Attack

One or more attackers accesses one or more social TAN links very fast. The objective of the attacker(s) is to overload the social TAN service.

To prevent this, the social TAN service needs to identify and discard the connection from the attacker(s) as early as possible at best, before the server starts to process the query.

This could be realized through a blacklist or by limiting the amount of connections per IP. Reactions can be, to discard the connection completely, to slow down the request, or redirect the request to an error website.

6 Conclusion and outlook

The social TAN service will enable privacy-aware people to keep in touch with people they meet without letting a third party discover their private social networking profile.

However, this is just a way to avoid the creation of linkability. It will not enhance the security of a social network. Nevertheless, it enables the publishing user to set the privacy policy to a maximum without losing the ability to being found by her contacts.

A possible future enhancement would be an implementation of the social TAN service in the social networks directly.

ISBN 978-3-88579-286-4

References

[APK+11] Demetris Antoniades, Iasonas Polakis, Georgios Kontaxis, Elias Athanasopoulos, Sotiris Ioannidis, Evangelos P. Markatos, and Thomas Karagiannis. we.b: the web of short urls. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 715–724, New York, NY, USA, 2011. ACM.

[BMHH+11] Michael S. Bernstein, Andrés Monroy-Hernández, Drew Harry, Paul André, Katarina Panovich, and Greg Vargas. 4chan and /b/: An Analysis of Anonymity and Ephemerality in a Large Online Community. *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[Eck10] Peter Eckersley. How Unique Is Your Web Browser? In Mikhail Atallah and Nicholas Hopper, editors, *Privacy Enhancing Technologies*, volume 6205 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin / Heidelberg, 2010.

[FHBH+99] John Franks, Phillip M. Hallam-Baker, Jeffery L. Hostetler, Scott D. Lawrence, Paul J. Leach, Ari Luotonen, and Lawrence C. Stewart. HTTP Authentication: Basic and Digest Access Authentication. RFC 2617 (Draft Standard), jun 1999.

[Hal94] Neil Haller. The S/KEY One-Time Password System. In *Proceedings of the Internet Society Symposium on Network and Distributed Systems*, pages 151–157, 1994.

[Ham50] Richard Wesley Hamming. Error Detecting and Error Correcting Codes. *The Bell System Technical Journal*, 26(2):147–160, 1950.

[LM10] Avinash Lakshman and Prashant Malik. Cassandra: a decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, 44:35–40, April 2010.

[LMS05] Paul J. Leach, Michael Mealling, and Rich Salz. RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace, 2005.

[Mil56] George Armitage Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

ISBN 978-3-88579-286-4