Information and Password Attacks on Social Networks: An Argument for Cryptography

Enrico Franchi, Agostino Poggi and Michele Tomaiuolo
Department of Information Engineering, University of Parma, Italy

Published in: Journal of Information Technology Research (JITR): 8(1)

ABSTRACT
Online social networks have changed the way people interact, allowing them to stay in touch with their acquaintances, reconnect with old friends, and establish new relationships with other people based on hobbies, interests, and friendship circles. Unfortunately, the regrettable concurrence of the users' carefree attitude in sharing information, the often sub-par security measures from the part of the system operators and, eventually, the high value of the published information make online social networks an interesting target for crackers and scammers alike. The information contained can be used to trigger attacks to even more sensible targets and the ultimate goal of sociability shared by the users allows sophisticated forms of social engineering inside the system. This work reviews some typical social attacks that are conducted on social networking systems, carrying real-world examples of such violations and analysing in particular the weakness of password mechanisms. It then presents some solutions that could improve the overall security of the systems.

Keywords: social networks; information security; cryptography; password cracking; social engineering.

INTRODUCTION
If we have to chose among the innovations of the past decade just a single phenomenon because of its outstanding social impact, that would be the diffusion of online social networks. While some social networking services were already active in the nineties, the capillary diffusion and the sheer number of people involved transformed online social networking in an unprecedented revolution only recently. Online social networks have already modified the way people interact. They allow users to reconnect with old friends, or to establish new connections with unknown people. In general, users are facilitated in maintaining and developing the relationships with their acquaintances, on the basis of common activities, interests, and contacts.
From a technological perspective, online social networks are mostly based on sets of web-based services that allow people to present themselves through a profile, to establish connections with other users in the system, and to publish resources. Moreover, these systems use common interests and the natural transitivity of some human relationships to suggest new contacts with whom to establish a connection. Although some of these aspects already appeared in other systems, online social networks represent an unprecedented cultural phenomenon. It is mainly characterised by the unceasing flow of information that users publish in such systems, and their relentless strife to increase the number of their virtual friends and acquaintances.
Unfortunately, online social networks are becoming an interesting target for crackers and scammers alike. In fact, many factors concur to attract malicious actions: (*i*) the users' carefree attitude in sharing information, (*ii*) the often sub-par security measures from the part of the system operators, and (*iii*) the high value of the published information. In particular, the information available through social media can be used to trigger attacks to even more sensible targets. Various forms of social engineering inside

the system, including sophisticated and long term attacks, may be facilitated by the general sense of sociability shared by users, which *per se* is an intrinsic objective of social media. However, while the only lasting solution to privacy and security issues would be increasing the users awareness, much can and shall be done at the system level in order to protect the data with cryptography and to decrease the impact of wrong choices and mistakes on the user's part.
This work reviews some typical social attacks that are conducted on social networking systems, carrying real-world examples of such violations and analysing in particular the weakness of password mechanisms. It then presents some solutions that could improve the overall security of the systems.

SECURITY THREATS ASSOCIATED WITH SOCIAL MEDIA

Nowadays, online social networks involve people from the entire world, of any age and with any kind of education. They also helped to increase computer usage among categories that previously showed little interest for it (Stroud, 2008). The users of information systems have various types of security requirements, including: *confidentiality*, *integrity*, *accountability*, *availability* and *anonymity*. The same security requirements can be applied to social networking platforms, as well.
Unfortunately, while most users are aware that their profile and the information they publish is essentially public, they usually strengthen their privacy settings only after problems arise and tend to overlook the actual impact of the information they disclose (Stroud, 2008). Apparently harmless information can be exploited, and the more information the attacker has, the more severe and sophisticated the attack can be. For example, name, location and age can be used to connect a profile to a real-world identity for more than half of the residents in the USA (Irani et al., 2011).
In fact, social networking platforms are susceptible to different types of attacks, targeting different components, conducted from different domains, using different techniques. For better analysing these attacks, it is useful to identify the main abstract components of a generic social networking platform, corresponding to different functional aspects of those systems. Attackers can target each of the different components, or they can target different levels, possibly with roughly the same logic. We identify four main components:

1. The *social networking* component. It manages and protects access to the users' personal profiles and the social relationships among users.
2. The *content management* component. It manages and protects access to all user generated content, including personal status updates, comments, links to other content, photos and multimedia galleries.
3. The *infrastructure services* component. It provides the basic infrastructure services needed to run the social networking platform, including storage and replication services for content and profiles, information indexing and routing, management of users' online presence.
4. The *communication and transport* component. It encapsulates basic inter-networking and ad-hoc networking functionalities.

Moreover, we can distinguish two different kinds of attackers:

1. *Intruders.* An attack can be conducted by users accessing the system without proper authorization, or with accounts created for conducting the attack, purposely.
2. *Insiders.* Also legitimate users or entities participating in the systems operations can assume malicious behaviours. From the users point of view, malicious behaviour can also be attributed to the service provider.

All the various classical kinds of security attacks may be adapted and applied to online social networks, too. The most typical threats against OSNs include:

- *Unauthorized Access.* Users who have not been granted adequate permissions for accessing some services and resources, may attempt to circumvent the security mechanisms and policies of the system and gain unauthorised access. In a social networking platform, any user who has access to some profiles and messages can harm their legitimate owners. The collection of existing data is the basis of profiling attacks. These data may also supply some knowledge for secondary data collection from a wide range of different sources, including other OSNs. Remote access can also occur at system level. In this case the attacker may directly gain control of all resources.
- *Social engineering.* In a social networking application, a common attack is to psychologically manipulate a user into performing misguided actions. It is similar to a confidence trick or a traditional fraud, but by means of computer-based communications and online social networking, typically to gain access to confidential information. In a phishing scheme, the attacker masquerades as a trustworthy entity to obtain the desired information. In most cases the victim and the attacker never acknowledged each other directly in real life.
- *Masquerading.* When a rogue user disguises his identity and claims the identity of another user, the former is said to be masquerading. Masquerading may be attempted by an attacker either during a conversation or while registering his own profile, for deceiving other users or the whole social networking platform. Sometimes, masquerading is the first step to gain access to infrastructure services and resources to which the attacker is not entitled. Simple impersonation, by cloning the victim's profile from the same platform or by porting profile data from a different platform, may easily lead the attacker to gain trust from the victim's contacts. This way, it can damage other users eventually deceived. Especially in communities where reputation is valued, masquerading can also damage the user whose identity has been stolen. In fact, the attacker may pretend to be another user in order to shift the blame for any liable action.

We will provide some examples and a discussion of these threats in the following subsections. However, also other traditional security attacks can be conducted against OSNs, including:

- *Denial of Service (DoS).* The services and communications at the infrastructure level can be disrupted by common denial of service attacks. Social networking platforms are also susceptible to all the conventional denial of service attacks aimed at the underlying operating system or communication protocols. In addition to attacking the whole infrastructure of a social networking platform, users can also launch denial of service attacks against specific users, especially in a distributed platform. For example, repeatedly sending messages or other spam may place undue burden on the recipient users and their systems. Malicious users can also intentionally distribute false or useless information to prevent other users from completing their social activities.
- *Repudiation.* In general, repudiation occurs when a user, after having performed some action, later denies that action having happened (at least under his responsibility). Repudiation can be intentional or even accidental. It can also be the result of a misunderstanding, when users have a different view of events. In any case it can generate important disputes. In a sense, nothing can prevent a user from repudiating one of his actions. But a social networking platform can eventually help resolving disputes by providing needed evidence, if it maintains a sufficiently

detailed log of events. For users who value their reputation, the availability of such evidence may constitute a valid deterrent.

- *Eavesdropping.* The attempt to observe the flow and possibly the content of confidential messages is one of the most classical security threats. Apart from reading the content of messages, which may require cryptanalysis, an eavesdropper may gather useful information by simply observing the pattern of messages and their recipients, for example inferring the type of services being requested. To eavesdrop on other users, an attacker may also exploit the infrastructure and communication services of the platform, e.g. through unauthorized access.
- *Alteration.* When a user signs up a social networking service, he starts exposing his profile and content to the platform. An attacker may tamper with the profile and content data published by the victim, with all the messages he communicates to other users and all data used on the infrastructure services. Alteration can also be conducted by the service operator, which provides the facilities for online social networking and may take control of published data. Alteration may take the particular form of filtering, or censorship, when applied systematically for removing undesired content from the OSN.
- *Copy and Replay.* Each action in a social network may be subject to copy and reply. In this type of security threat, an attacker attempts to intercept some data and clone it, for retransmitting it later. The interceptor may successfully copy and replay a message, a complete profile or any other data. If those data are not associated with a signature and a timestamp, the repeated reception of such copies may pass unnoticed and accepted as a legitimate action.

Types of information leaks from OSNs

According to Li, Li and Venkatasubramanian (2007), there are two types of privacy attacks in online social networks:

1. *Identity disclosure.* It occurs when the adversary is able to determine the mapping from an anonymous profile to a specific real-world entity (e.g. an individual).
2. *Attribute disclosure.* It occurs when an adversary is able to determine the value of a user attribute that the user intended to stay private.

Moreover, attackers may gather data from:

1. *A single social networking platform.*
2. *Multiple platforms, services or extension apps.*

Sensible data may be obtained both through:

1. *Direct access*, either through an authorized service or through some breach, and
2. *Attribute inference*, based on public data available about a user's contacts, and the correlation usually existing among attributes of linked profiles.

Either about identity or attributes, privacy in online social networks is mostly intended as user-to-user privacy: even when the relative settings are set correctly, so that no other user in the system can access information not intended for his eyes, the system itself has full access to information. In fact, in most online social networks, the system owners actually rely on such information to make a profit, for example to improve the accuracy of target advertisement. Unfortunately, as long as they have full

access to the information – i.e., the information is available in the system in the clear – any security issue or naivety results in privacy violations.
Even innocuous features can be easily become serious issues. For example, in order to promote the service, Facebook had a public directory containing the names of the users that used to be presented along with 10 random friends of theirs. Such feature allowed web-spiders to repeatedly request pages from such directories and essentially discover the structure of the network. Subsequently, the number of friends was reduced to 8 and the selection of friends became a deterministic function of the IP address of the requester. However, even the greatly reduced information can still be used to violate privacy (Bonneau, 2009).
Another very relevant problem is the amount of integration we expect from digital services. Online social networks are relatively open system that can be enhanced by third party widgets and games, usually called apps, which are embedded in the user's pages and typically have access to some amount of user information. Moreover, the users often want different services to interoperate for various reasons, so that, for example, a new tweet is also notified to the Facebook friends or pictures stored on Flickr are accessed by an online printing service. Eventually, the credentials of the online social network can be used, in many cases, to log in a different system without needing a separate account. The problem is that when any of these external services, systems or apps is violated, private information can be stolen. For example, in June 2012, although the main Twitter servers were not compromised, sensible data was nonetheless stolen from a third party widget, TweetGif (Robertson, 2012).
Similarly, we expect our mobile devices to interact cleverly with the online social networks. Service providers create mobile applications, which, in turn, have some degree of access to the data stored on the mobile device. Although such applications usually do not maliciously access user's data for purposes different from those stated, sometimes privacy is neglected. For example, the iPhone LinkedIn App used to send all the user's calendars to central LinkedIn servers, including phone numbers, call details and passcodes, while only the relevant information should have been sent (Cheng, 2012).
Even when the social networking platform does not provide data directly, yet many supposedly private properties may be inferred about a user. In fact, in social networks the attributes of users who share some kind of link are often correlated. Zheleva and Getoor (2009) introduce different models of attacks, to infer the hidden sensitive values on the basis of friendship links or public group membership data. Specifically:

- *Friend-aggregate.* It looks at the sensitive attribute distribution amongst the friends of the person under question.
- *Collective classification.* It aims at inferring class labels of linked objects together, instead of classifying each instance independently of the rest. It iteratively uses inferred values for connected private profiles, in addition to public ones.
- *Flat-link.* It deals with links by flattening the data and considering the adjacency matrix of the graph. Each user has a list of features, corresponding to the size of the network. The public profiles are used as a training set for the classifier.
- *Blockmodelling.* It is a stochastic model, supposing that users form natural clusters or blocks. Then, their interactions can be explained by the blocks they belong to.
- *Groupmate-link.* In this model, groupmates are considered as friends to whom users are implicitly linked. It assumes that each group is a clique of friends. Thus, it creates a friendship link between users who belong to at least one group together, without representing the strength

of the link. The resulting network can then be analysed using one of the previous models.

- *Group-based classification.* It considers each group as a feature in a classifier, and sensitive attributes are inferred from the groups a user belongs to.
- *Basic.* It calculates the overall marginal distribution of public data and uses it to infer sensitive attributes of private profiles. This is the simplest model, which can be applied also in the absence of relationship and group information.

Phishing and impersonation attacks

Social engineering has always been a major security threat to information systems. Two of the more frequent kinds of social engineering attacks, i.e., pretexting and phishing, require some background information, but such information was not always easy to obtain. In fact, some of the early social engineering attacks actually involved going through company or people's trash bins in order to find scraps of paper containing relevant information. Nowadays, many interesting pieces of information are publicly available in social networking systems or accessible only to victim's friends at best (or at worse, from the attacker's perspective).

Traditional phishing consisted in sending electronic communications to a huge number of individuals with the intent to have them disclose some relevant information, such as passwords or credit card numbers. Since the large scale of such attacks, even if only few users actually trusted the communication and disclosed information, it was nonetheless enough to make the attack profitable. Such communications typically used very generic ways to address the victim and typically avoided any specific detail so that the same message could be sent to all the recipients. Nowadays, people have been instructed not to trust such generic messages, and other forms of phishing were created.

Spear phishing is a form of phishing where the attack is directed against a single individual and the offending communication contains large amounts of information on the victim himself, in order to convince him of the communication authenticity. This strategy is very effective both if the information employed is public but the victim somewhat thinks it is not or if the attackers gained access to private information by other means. Another popular form of phishing is cloning: a cloning attack requires an original legitimate message that is subsequently tampered and sent the victims. These communications look extremely authentic, but nonetheless they contain ways to direct the victims towards the attackers website (where he is usually requested his credentials). Data-leaks offer an even more dangerous variant: when the data leak becomes public, users expect their service providers to send them emails notifying them to change their password. If such mails are instead well-crafted phishing emails, that perhaps also use some private information about the user, many more people fall for the scam. The kind of attack just described actually happened in the days immediately after the LinkedIn breach.

A serious weakness in many online services regards the "security questions". In fact, while phishing can be avoided by paying attention to the actual identity of the communications initiators, not much can be done against "security questions" attacks. Some services allow the users to choose the security questions so that the information required to answer it correctly is not publicly available, however, for many other sites only a closed list of security questions is available. Researchers estimated that more than 33% of the security questions involve names of relatives or friends and about 16% involve personal preferences ("favorite something"). Popular security questions are also the name of a pet (typically the current or first one), ZIP code or social security number (Rabkin, 2008). The victim in online social networks often directly or indirectly divulges such pieces of information. Even apparently safer information, such as social security numbers, may be guessed with high confidence from publicly available data (Acquisti & Gross, 2009).

This kind of attacks is usually not directed towards online social networks, at least initially, but usually

targets email service providers, because, when the security question is answered correctly, most systems send a password-reset email, and, consequently, controlling the main email account is necessarily the first step. Similarly to spear phishing, security-question related attacks target only specific individuals. Popular examples are the violation of Sarah Palin's email account in 2008, using information publicly available on Wikipedia and on official websites and, more recently, the attack against Mat Honan. In the latter case, the tech journalist had his mail, iCloud and Twitter accounts compromised after an elaborate chain of social attacks, during which also his mobile phone, tablet and laptop have been remotely wiped (Honan, 2012).

Another typical attack on social networking systems is impersonation (identity theft), which can occur either on a site where the victim already has a profile (profile cloning) or on different site (cross-site profile cloning). In both cases, it is easy to convince the victim's acquaintances to accept friendship from the fake profile and subsequently to disclose confidential data. Experiments on the effectiveness of impersonation have been conducted and the results show that such attacks mostly succeed (over 50% success rate) even in the case of simple non-cross-profile cloning (Bilge et al., 2009).

When the attackers obtain credential for an online social network profile, additional attacks can be performed. If the attackers are interested in keeping a private database containing the violated accounts data, they can create a widget that copies the relevant resources or simply scrapes the web pages. Moreover, they can try to use the same credential to attack the user's accounts in other online services. Since users do not often use different passwords for different accounts, this strategy is rather effective. For example, during a period of roughly a week in June 2012, several services were compromised. About a month after this so-called "breach week", a number of Dropbox accounts have been hacked, using the passwords stolen from the already compromised accounts (Agarwal, 2012).

Another possibility is using the violated account to accept friendships from profiles controlled by the attackers. If the victim has already enough friends (on average, a Facebook user has 1000 friends), it is not likely that he notices the new friendship, especially because the attackers can remove any notification before the account owner actually sees that. Such friendship can be subsequently used as a Trojan horse to access the user's data even after the user changes the password and, moreover, is also likely to attract friendship from the victim contacts, considering that mutual friendships usually increases the confidence that an accounts should be trusted.

Controlling many profiles in a social network, either real ones or fake, can be used for spam purposes, to influence opinions in general or simply to gather huge amounts of information. In fact, in order to either access the data of most users in the network or deliver them a message appearing to come from a friend, it is not necessary to control a large number of accounts, but it is sufficient to control accounts that approximate a dominating set, i.e. a set D such that $V = D \cup friends(D)$, where V is the set of all users in the network. Although computing a minimal dominating set is an NP-complete problem and requires full knowledge of the networks, decent approximations can be made using greedy algorithms and using only sampled knowledge.

THE WEAKNESS OF PASSWORDS

Most of the attacks we discussed so far are forms of social engineering. However, online social networks are not, from a technical point of view, different from any other password-protected service: if the passwords are not chosen judiciously and stored in a safe way, then the violation of accounts becomes very easy. In fact, the analysis of available records about the robustness of users' passwords, as well as the security mechanisms and policies deployed in online systems, does not shape a reassuring scenario. Table 1 summarizes some major password leaks suffered by social websites in recent years.

Date	Target site	Size of leak	Password storage
2012, June	LinkedIn	6,5 million passwords	SHA1, not salted
2012, June	eHarmony	1,5 million passwords	MD5, not salted
2012, June	last.fm	2,5 million passwords	MD5, not salted
2012, June	League of Legends	32 million passwords	MD5, not salted
2011	PlayStation Network	77 million accounts	Hashed, not salted
2011	Sony Online Entertainment	25 million accounts	Hashed, not salted
2009	Rockyou	32 million accounts	Clear text

Table 1. Major password leaks

The "Breach Week"

The first week of June 2012 has earned the fame of "Breach Week". Some million passwords were stolen, in order, from LinkedIn, eHarmony and last.fm accounts. At least the first two incidents appear correlated, as the passwords were published by the same user on a Russian forum about cracking. The first file, leaked from the professional social site LinkedIn, totalled 6,5 million unique passwords. It is not clear if other passwords were leaked, but the episode may have involved many more users. The second file, leaked from the popular online dating site eHarmony, contained around 1,5 million passwords. Finally, the internet radio platform last.fm notified all its users of a possible leak of passwords, requesting to update their login information. Administrators announced ongoing investigations. A published file accounts for at least 2,5 million passwords leaked form the radio platform. But these episodes are not isolated. In fact, few days later, an alert appeared on the online gaming platform League of Legends, notifying its 32 million users of a breach in some of its databases (Merrill & Beck, 2012). The site administrators revealed that more than half of the passwords would not resist to crack: "*We compared encrypted password hashes and discovered that 11 passwords were shared by over 10,000 players each... A double-digit percentage of individuals had the same password as at least one other person.*" Apart from revealing a widespread lack of awareness among users about basic password security, the announcement may imply that passwords, like on LinkedIn and eHarmony, were hashed but not salted (Yin, 2012).

Sony, in 2011, suffered from breaches in its PlayStation Network and Qriocity media streaming service. The services were suspended for almost a month. Personal data from around 77 million accounts were stolen, possibly including credit card numbers. During the same period, Sony Online Entertainment announced that another breach led to leaking personal data from 25 million accounts. These breaches compare to previous massive data leaks from TJX Companies, a fashion retail network, in 2007, affecting over 45 million customers, including details about credit card and in some cases also social security numbers and driver's license numbers. In 2009, Heartland Payment Systems, currently the fifth largest credit card processor in the United States and the 9th in the world, was affected by a breach which costed something in the tens of million dollars. In 2005, a similar breach at CardSystems Solutions, another payment card processor, jeopardized roughly 40 million credit and debit card accounts (Vijayan, 2007).

One of the best known episodes, finally, is the breach suffered by Rockyou. After an SQL injection attack in 2009, the intruders gained access to the website database, which included the full list of unprotected clear text passwords of all 32 million users. In fact, those passwords are still readily

available over the BitTorrent network and are being used for dictionary attacks against other websites (Cubrilovic, 2009).

Common password storing and cracking techniques

The safe management of passwords is an old problem. Various password storing mechanisms can be considered.

1. *Clear text.* Though used in practice, the storage of passwords as clear text should out of the question, as it offers no protection against intruders.
2. *Encryption.* The use of traditional encryption schemes is also discouraged. In fact, by knowledge of the decryption key, all passwords may be subverted in a single shot. If an intruder acquires the control of a machine, then the possibility of loosing a decryption key is quite concrete.
3. *Hashing.* The solution adopted since decades in Unix systems is based on cryptographic one-way functions, that can only be inverted by guessing the original clear text password (Morris & Thompson, 1979). However, common hashing algorithms are often designed for efficiency, which allows attackers to try many combinations in short time. Moreover, the effort to guess users' passwords can be reduced by attackers, if they generate the hash of a tentative password and confront it with each one of the actual password hashes of the attacked system.
4. *Salting.* If some unique value (a salt) is added to each password before hashing it, the result is unique for each user. If two users use the same password, two different hashes are obtained, since that password is combined with two different salts. Then, in the database, both the hash and the salt, in the clear, need to be stored. Thus, it is not possible to pre-compute hashes for all popular and simple passwords, or for all combinations generated through brute force (Morris & Thompson, 1979).
5. *Password hashing algorithms.* While common hashing algorithms are designed to be as fast and efficient as possible, password hashing algorithms are designed to require a significant amount of computational resources. Bcrypt, one of the best options among password hashing algorithms, is based on the Blowfish algorithm and allows developers to decide the number of iterations of its main function, possibly requiring various orders of magnitude more time than generic hashing algorithms. The exact choice depends on the desired balance of password security and needed computational resources for normal operation, in particular for handling the regular number of logins (Provos & Mazieres, 1999).

Though password storing mechanisms are well known and documented, they are not always used in existing systems, including some popular services, with large user bases. In fact, some lessons can be learned about implemented mechanisms for password security in real cases.

A number of sites adopts techniques that are far from the best practices in this field. We will leave the Rockyou case apart. LinkedIn, for example, avoided storing passwords in clear text, but used a suboptimal algorithm for hashing. In fact, it used a good generic hashing algorithm (SHA-1, namely), instead of a password hashing algorithm, like bcrypt. On moderate hardware, SHA-1 can be computed over almost 200MBs of data per second, and MD5 over more than 300 MB of data per second (Dai, 2009). With these algorithms, a password of 6 lowercase alphanumeric characters can be easily obtained through a brute force attack in less than a minute. And this is without using the potential of parallel GPU computing, which can obtain results which are at least an order of magnitude better. Exploiting four HD 5970 cards and some precalculations for the latest steps of MD5, the Whitepixel tool may achieve 33.1 billions MD5 hash/s, on a system costing 2.700 $ at the end of 2010 (Bevand,

2010).
Another lesson that can be learned is that many websites simply skip password salting, even if it is a well established technique (Morris & Thompson, 1979). LinkedIn and eHarmony are not isolated examples, though emblematic given their huge user bases. For example, it took many years and versions for the popular blogging platform Wordpress to finally add salt to its user passwords, in 2008 at version 2.5.
In all those careless sites, simple attacks can be based on *dictionaries* of popular passwords, together with *mangling rules* to obtain similar and derived passwords. Another possibility is to try all possible combinations of lowercase letters, uppercase letters, digits and punctuation symbols, in a *brute force* attack. Some tools, just like John the Ripper, can apply both attacks on a given list of hashed passwords. Starting from a dictionary or a combinatorial engine, the obtained password is hashed and then compared to all available hashes, possibly leading to the discovery of one or more users' passwords after a single hash operation. The effectiveness of the operation is greatly simplified by the fact that a single algorithm is applied against all passwords, without salt or additional parameters. Moreover, if passwords are not salted, the attacks can be made even more effective by calculating in advance the hashes of all possible passwords, up to a certain length. Obviously, taking into account the needed disk space, this approach is feasible only for very short passwords. But techniques are available to trade time for space, thus reducing the needed disk space but requiring more hash calculations at runtime. Among such techniques, some are based on the so-called *rainbow tables*. Oechslin (2003) shows how a previous technique, described by Hellman and refined by Rivest, could be further improved, halving the number of calculations during cryptanalysis.
Those methods are all based on the iterative calculation of a hash function and a reduction function, in an alternating sequence, starting from a given password and repeating the cycle some thousands of times, depending on the desired balance between space and runtime processing time. For a given chain, only the starting password and the final hash are stored, while intermediate results are discarded. The number of chains to store depends on the desired success probability in decrypting a given hashed password. In the original paper, the method is applied to Windows LanManager passwords. With a space of 1.4GB for rainbow tables (and thanks to the weakness of the old LanManager scheme) a success rate of 99.9% can be achieved.
Given a certain hash, finding the corresponding password requires finding a rainbow chain in the table. If the original hash is not found, then one or more cycle of the reduction function and hash function are applied and then the search is repeated. Finally, when the relevant rainbow chain is found, starting from the first password in the chain, all calculations are repeated, till the password associated with the original hash is found.

ANALYSING SOME MILLION USERS' PASSWORDS

Unfortunately, users do not seem to choose passwords with care. A first glimpse of this fact comes from the security alert issued after the League of Legends intrusion (Merrill & Beck, 2012). In that alert, the system administrators pointed out that a high percentage of users had the same password of another user. The reported data of 10000 different users sharing the same 11 passwords, is a glaring evidence. So, in order to confirm the impression that users choose very weak passwords, we decided to analyse the publicly available list of leaked hashes from LinkedIn.

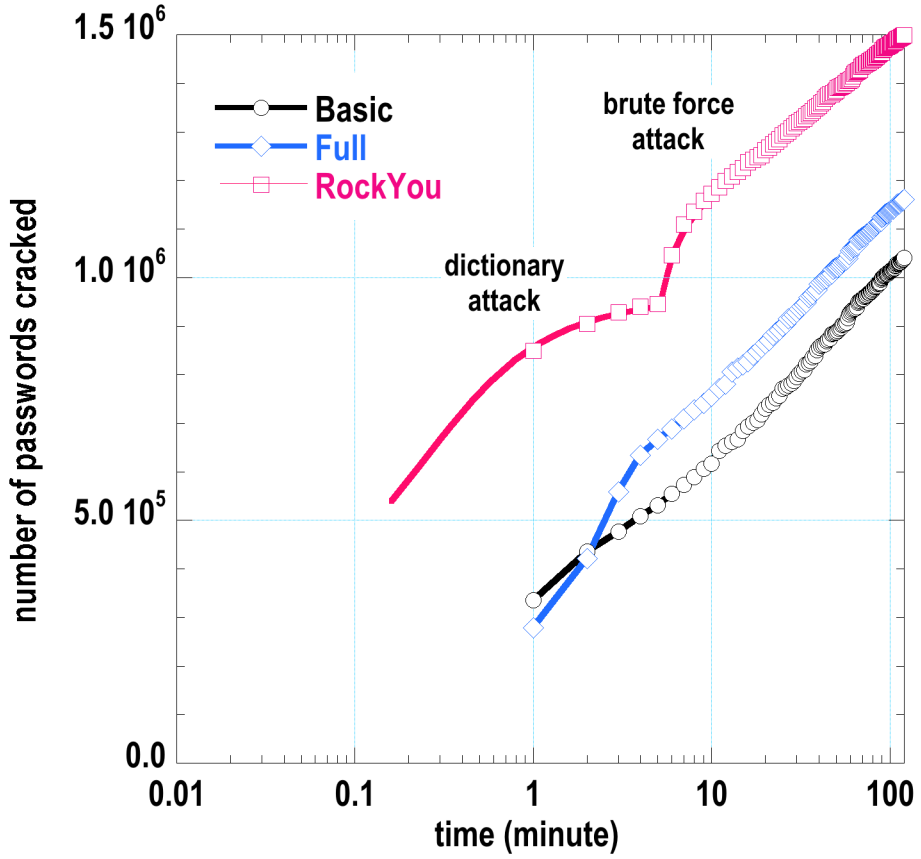

Figure 1. Dependence of the number cracked by using both dictionary (labeled as "Basic", "Full" and "RockYou") and brute force attacks, as functions of cracking time. In the result of the "RockYou" approach it is particularly evident how the "brute-force" attack starts after exhausting the "dictionary" attack.

The first step is to crack the hashes to obtain the original passwords. Since the hashing procedure is essentially irreversible, the typical strategy is to "guess" which string generated the hash. One possibility is to brute-force attack the hash, i.e., simply hashing every possible string and comparing it with the desired hash. Such strategy is very slow and ineffective in practice whereas dictionary attacks are much more effective. In a dictionary attack the strings are not chosen at random, but are extracted from a dictionary of common words. Additionally, the words from the dictionary are mangled simulating the usual strategies users employ to make passwords harder to guess (e.g., substitute numbers for alphabetic characters, appending values to the end or the beginning of the words). Although more techniques exist, we decided to start cracking the passwords using the popular password cracker "John the Ripper", which performs both brute force and dictionary attacks with mangling rules and we reported our results in Figure 1. All the attempts were performed on a Core2 Duo laptop at 2.40 MHz. Our first attempt ("Basic" in Figure 1) used a small dictionary of around 3500 common password that is distributed with John itself and we also enabled simple mangling rules for generating common derivative passwords, in two hours the program was able to recover more than a million (1.05 M) LinkedIn users' passwords. The second attempt ("Full" in Figure 1) used a much larger dictionary, containing a few million words from more than 20 languages, also available from the program website. The larger dictionary provides some marginal gain.
Eventually, we fed John with the database of 32 million passwords that were stolen from the RockYou

website in 2009 and subsequently published on the web. The results significantly improved. In two hours around 1.5 million passwords were revealed, 44% more than those obtained with the first attempts. In this case, the first million passwords were obtained in less than five minutes.
The most likely explanation is that the LinkedIn password set has a large overlap with the RockYou dataset, which, in turn, confirms the idea that many users tend to reuse passwords and are not extremely creative in deciding their own passwords. Moreover, in this case, of the 2 million passwords we cracked in 24 hours starting with the RockYou password dictionary, more than 25% are not longer than 6 characters and that less than 15% are longer than 8 characters. Moreover, almost the 30% consists only of alphabetic lowercase characters and another 30% consists of alphanumeric characters, with no special characters, and, eventually, almost 1 password over 10 consists of only digits. Roughly 12% of the passwords end with the digit 1, 200.000 passwords end with 123 and about half as many with 1234. Years are also frequently included (e.g., numbers from 1975 to 2011, while 2012 is significantly less popular and "future" years are rare). The most popular base word that constitutes the password is a variation of the name of the website ("linked", "linkedin" and "link") and "love" and "password" still made it in the top 10. Analyses of other leaks of passwords show that variations on the service name are popular and that "password" is an evergreen (Dragusin, 2012).
In fact, the passwords that have not been cracked during our 24h timeframe are expected to be much stronger (otherwise they would have been cracked as easily) and the fact that only 1/3 of the password was cracked is somewhat encouraging. It is also interesting that, using the RockYou dataset as dictionary, the number of password cracked ($N(p)$) depends on the length of the passwords ($L(p)$) according to $N(p) = A\, e^{-BL(p)}$ where $A = 3.15x106$ and $B = -0.786$.
The last consideration regards the nature of the dataset: repeated passwords were not included, i.e., in the original system more than one user could have the same passwords, so that 2 million passwords could account for far more than 2 million of users, and, considering the remarks of the operators of League of Legends, they probably do.

HOW TO MAKE ONLINE SOCIAL NETWORKS MORE SECURE?

The huge amount of information and the features-over-security approach of online social networks is a serious security problem, especially considering that some million of vulnerable accounts remain a big security threat for other systems as well, in the sense that, considering high password reuse, the more services are violated, the more likely it is that accounts on other systems are violated as well.
Many solutions can be used to increase security. While the best long-term solution to the security problems is increasing the users' awareness, this may be unpractical because of the huge amount of people involved and their perception to be in a relatively safe environment, where friends are just few clicks away. However, there are many solutions that providers and software developers can use in order to improve the overall security.
For, example, most browsers and email clients offer some features to automatically detect phishing and one of the major weaknesses, i.e., security questions, can be fixed either improving the quality of the questions themselves (Rabkin, 2008) or, better, sending reset messages via SMS, which at least would require the attacker to physically access the victim's mobile phone.
Zhou, Pei and Luk (2008) suggest some anonymization techniques to prevent privacy attacks in social networks. In particular, they suggest two different approaches, based respectively on generalization and perturbation:

1. *Clustering.* Vertices and edges are clustered into groups, so that a subgraph can be abstracted into a supervertex. This way, details about individuals are hidden.
2. *Graph modification.* Some vertices and edges of the graph are modified, by insertion or

deletion. Such modifications may be applied using an optimization approach, randomly, or greedily, to match privacy requirements.

Felt & Evans (2008) suggest to protect data of social networks, especially from exposure to third-party developers. They present a privacy-by-proxy design, which mask data presented to external applications, using placeholders instead of actual values. However, it is not clear how resistant this simple approach is against de-anonymization techniques.

Li et al. (2007) discuss such de-anonymization techniques, under the conditions of k-anonymity. Under those conditions, each node is undistinguishable from other k-1 nodes, with regards to attributes which may potentially identify an individual. This group of node constitutes a class of equivalence. While k-anonymity can protect against identity disclosure, it fails to protect against attribute disclosure in the general case. Another discussed notion of privacy is the l-diversity, which requires each class of equivalence to contain at least l different values for a sensitive attribute. Authors propose instead t-closeness as a measure of privacy, which requires that the distribution of an attribute in a class of equivalence is close enough to that of the whole network, being less than a threshold t.

Weak passwords remain a major problem. Although, some service providers are either increasing strictness of the conditions to accept a password as valid, experience shows that "lazy" users always circumvent syntactical conditions (Yan et al., 2004). For example, in order to make passwords more secure the providers started to require that at least one character should be a number. As a consequence, many users simply appended a single digit to their weak password, without neither substantially increasing the password security as a whole nor realising they are failing to do so.

Part of the problem is that, when choosing passwords, users are facing the choice between easy to remember, easy to type passwords and secure passwords. As expected, users tend to choose the former. Even if they are instructed with good techniques to build secure and memorable passwords (e.g., pass-phrases), the large amount of services they subscribe creates the problem to remember which password is needed for which account and, consequently, users tend to reuse their passwords over and over, with terrible security implications. Also when longer pass-phrases are chosen, users prefer easier to remember sequences, which make sense to them. But this motive also makes them vulnerable to more advanced dictionary attacks, which gather phrases from popular books and online sources (Goodin, 2013).

Many solutions exist to improve password security, i.e., password managers and biometric identification. However, these solutions have serious drawbacks. Biometric identification is cumbersome because it may require specialised hardware and is often perceived as a privacy violation on its own right. Additionally, biometric data, when used for authentication, are roughly equivalent to strong and long passwords, which are hard to guess but which, on the other hand, are impossible to change. This way, if a service requiring biometric data at login is broken, or behaves in a rogue way, then the security based on those biometric data, on all sites, is made ineffective or completely broken. Password managers, on the other hand, create a single point of failure that, if violated, compromises the security of every single account the user had. In fact, this is not just a remote hypothesis (Slattery, 2011).

All considered, we think that one of the best strategies to improve the security of online social networks would be using strong asymmetric cryptography to protect all the data stored inside the system. Decrypting data protected with state-of-the-art cryptography is very hard. The system can be designed so that data travels only in encrypted format and is decrypted only on the client machine, so that, in essence no secret is revealed to a third party during regular usage. However, regular asymmetric cryptography has high computational costs when messages have to be sent to a restricted audience that varies its members during time, as in the case of a "group" or "circle" of friends (Canetti et al., 1999).

The solution we favour is using flexible attribute-based encryption, such as CP-ABE (Bethencourt et al., 2007). With attribute-based encryption, it is possible to create subordinate key-pairs associated with arbitrary attributes. Such attributes can indicate belonging to some group or time-dependent access rights. The system operators and other users have not necessarily access to the data so that, even if the system is compromised, the data remains secure.

The main problem is that, without remotely storing the private key, it becomes difficult to access the online social network from multiple devices, such as from home and from the office. In the specific case of mobile-centric online social networks, it becomes possible to keep the private key on the mobile device without impairing functionality. Backup and remote wipe procedures are widely implemented and they can be used to solve most issues regarding losing the physical device.

Another positive effect of using flexible cryptography, such as attribute-based encryption, is that it becomes possible to temporarily grant rights to another device from the main mobile device where the private key is stored. In fact, this essentially allows using the mobile device as a non-intrusive hardware key without actually having to copy the private key around.

The essential problem with encrypting data is that usually the system operators want to access such data in order to improve target advertisement, that is their main source of revenue. Moreover, information quality, flow experience and trust in OSNs users' loyalty are correlated; thus, system operators have to pay close attention to them, to retain and increase their customers (Suki, 2012). So, although technically feasible, an end-to-end encryption strategy would require either an entirely different, although similarly profitable, business model or a less expensive architecture, typically obtained moving from centralized to distributed.

There are two main categories of distributed social networks: federated and peer-to-peer (P2P). In a federated system multiple entities cooperate to provide the service and each of them provides access to the whole system to a subset of the total users. Each user can choose the federated provider that he prefers, for example because he considers it worthier of trust. Service providers can also use high security and privacy standards as a way to attract more users and, perhaps even publish the code open source to increase confidence in their criteria. Direct access to the source code is possibly one of best way for user to discover whether their provider is using inadequate security standards, and, in general more competition and no lock-in are positive factors for increasing quality.

On the other hand, in a P2P system, every participant is both a user and a system provider at the same time. However, purely P2P system can have issues with data availability, i.e., post from very badly connected users can be difficult to obtain. Consequently P2P system may introduce “super-nodes” with a role akin to that of a federated provider. The distinction between the two categories is in practice blurred. The main advantage of P2P over federated approaches is that in the P2P scenario the code runs on the user machine and, potentially, no information is transmitted outside with the explicit user content. Moreover, trust management and negotiation mechanisms among users can be used in order to improve the availability of the data even without introducing super-nodes (Tomaiuolo, Poggi & Franchi, 2013).

Regardless of the actual form of distribution, the operative costs are essentially shared among multiple entities, and, in the case of P2P the required resources are already provided by the users, so that most costs are already covered.

Specifically in the field of social networking, various systems are being developed on the basis of peer-to-peer communications and DHT indexing, including Safebook (Cutillo, Molva & Strufe, 2009), PeerSoN (Bodriagov & Buchegger, 2013) and Blogracy (Franchi, Poggi & Tomaiuolo, 2013). Safebook is based on a network of socially close peers, defined Matryoshka. Peers in a user's Matryoshka are trusted and support the user by anonymizing communications and replicating content and profile information. PeerSoN proposes a Broadcast Encryption protocol, where each recipient has a different

key, which can be used to decrypt data received by the broadcaster. Blogracy, on the other hand, adopts an Attribute-Based Encryption protocol for protecting access to users' content. It allows each user to assign credentials to various groups of followers, for accessing protected content.
In conclusion, it should be noted that cryptography does not outright prevent “mass” phishing, however it can make much more troublesome to discover enough data on users to conduct credible spear-phishing attacks. Cloning phishing attacks are also less likely to succeed in federated networks, since different federated providers would typically use different standards, and become meaningless in pure P2P scenarios. Similarly, impersonation attacks cannot be directly avoided, but cryptography still makes it harder to gather enough information to successfully conduct the attack. Another partial defence against impersonation is that the same cryptographic keys that are used to guarantee confidentiality can be used to guarantee authenticity, i.e., any message and any piece of information can be guaranteed to come from the owner of the key: a clone could be easily detected.
Security questions are used in order to allow users to change their password in case it is stolen. Depending on how the system is designed, there may not be a password to reset altogether. In these cases, a similar problem is losing the private key. Although, in theory, backups should avoid the problem, a failsafe mechanism may still be useful; however, because of the way the system works, it should probably be something completely different, hopefully not vulnerable to the classic problems of “security questions”. Moreover, in P2P system there is not even the concept of a central authority that could “reset” any password or secret for the user and web of trust or similar strategies should be employed.
Essentially, since all the pieces of information are encrypted, data published in online social networks cannot be used to gather information to violate other systems. Moreover, even if the online social network services were violated, the attackers would not access any data in clear.

CONCLUSION

In this work, we reviewed some typical social attacks that are conducted specifically on social networking systems. In fact, along with the large number of legitimate users, these systems are also attracting the interest of crackers and scammers, who may seek information for triggering attacks to even more sensible targets. Real-world examples of such violations are already available. Actually, many factors concur to attract malicious actions: (*i*) the users’ carefree attitude in sharing information, (*ii*) the often inadequate security measures from the part of the system operators, and (*iii*) the high value of the published information.
While the only lasting solution to privacy and security issues would be increasing the users' awareness, much can and shall be done at the system level in order to protect the data with cryptography and to decrease the impact of wrong choices and mistakes on the user’s part.
Security is a complex issue, especially considering that most online social network users are not willing to be proactively engaged in the process. As a consequence, the system designer should pay additional care so that most interactions have high security standards regardless of the user understanding of the problem. However, for most online social networks providers’ security does not appear to be a priority, possibly because it is a feature that does not help to “sell” the product. Unfortunately, every system that is exploited makes every other system a bit less secure, because information leaks make information attacks more likely to succeed. Moreover, security is going to be an increasingly important problem in the future.
Therefore, we think that there is an expanding niche for smaller service providers willing to invest in security and that they can actually enter the market using decentralised architectures in order to decrease the costs of running the services. Such services could be designed from scratch in order to provide good security standards, possibly using attribute-based encryption.

REFERENCES

Acquisti, A., & Gross, R. (2009). Predicting Social Security numbers from public data. *Proceedings of the National academy of sciences*, *106*(27), 10975-10980.

Agarwal, A. (2012). *Security update & new features.* Retrieved January 10, 2014, from https://blog.dropbox.com/bre/07/security-update-new-features/

Bethencourt, J., Sahai, A., & Waters, B. (2007, May). Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP'07. IEEE Symposium on* (pp. 321-334). IEEE.

Bevand, M. (2010). *Whitepixel.* Retrieved January 10, 2014, from http://whitepixel.zorinaq.com/

Bilge, L., Strufe, T., Balzarotti, D., & Kirda, E. (2009, April). All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web* (pp. 551-560). ACM.

Bodriagov, O., & Buchegger, S. (2013). Encryption for peer-to-peer social networks. In *Security and Privacy in Social Networks* (pp. 47-65). Springer New York.

Bonneau, J., Anderson, J., Anderson, R., & Stajano, F. (2009, March). Eight friends are enough: social graph approximation via public listings. In *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems* (pp. 13-18). ACM.

Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., & Pinkas, B. (1999, March). Multicast security: A taxonomy and some efficient constructions. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Vol. 2, pp. 708-716). IEEE.

Cheng, J. (2012). *Your iPhone calendar isn't private—at least if you use the LinkedIn app*. Retrieved January 10, 2014, from http://arstechnica.com/apple/2012/06/your-iphone-calendar-isnt-privateat-least-if-you-use-the-linkedin-app/

Cubrilovic, N. (2009). *RockYou Hack: From Bad To Worse*. Retrieved January 10, 2014, from http://techcrunch.com/2009/12/14/rockyou-hack-security-myspace-facebook-passwords/

Cutillo, L. A., Molva, R., & Strufe, T. (2009). Safebook: A privacy-preserving online social network leveraging on real-life trust. *Communications Magazine, IEEE*, *47*(12), 94-101.

Dai, W. (2009). *Crypto++ 5.6.0 Benchmarks*. Retrieved January 10, 2014, from http://www.cryptopp.com/benchmarks-amd64.html

Dragusin, R. (2012). *Data breach at IEEE.org: 100k plaintext passwords.* Retrieved January 10, 2014, from http://ieeelog.com/

Felt, A., & Evans, D. (2008). Privacy protection for social networking APIs. *2008 Web 2.0 Security and Privacy (W2SP'08)*.

Franchi, E., Poggi, A., & Tomaiuolo, M. (2013). Open social networking for online collaboration. *International Journal of e-Collaboration (IJeC)*, *9*(3), 50-68.

Goodin, D. (2013). *How the Bible and YouTube are fueling the next frontier of password cracking*. Retrieved January 10, 2014, from http://arstechnica.com/security/2013/10/how-the-bible-and-youtube-are-fueling-the-next-frontier-of-password-cracking/

Honan, M. (2012). *How Apple and Amazon Security Flaws Led to My Epic Hacking.* Retrieved January

10, 2014, from http://www.wired.com/gadgetlab/2012/08/apple-amazon-mat-honan-hacking

Irani, D., Webb, S., Pu, C., & Li, K. (2011). Modeling unintended personal-information leakage from multiple online social networks. *Internet Computing, IEEE*, *15*(3), 13-19.

Li, N., Li, T., & Venkatasubramanian, S. (2007, April). t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on* (pp. 106-115). IEEE.

Merrill, M., & Beck, B. (2012). *League of Legends Account Security Alert.* Retrieved January 10, 2014, from http://euw.leagueoflegends.com/news/league-legends-account-security-alert

Morris, R., & Thompson, K. (1979). Password security: A case history. *Communications of the ACM*, *22*(11), 594-597.

Oechslin, P. (2003). Making a faster cryptanalytic time-memory trade-off. In *Advances in Cryptology-CRYPTO 2003* (pp. 617-630). Springer Berlin Heidelberg.

Provos, N., & Mazieres, D. (1999, June). A Future-Adaptable Password Scheme. In *USENIX Annual Technical Conference, FREENIX Track* (pp. 81-91).

Rabkin, A. (2008, July). Personal knowledge questions for fallback authentication: Security questions in the era of Facebook. In *Proceedings of the 4th symposium on Usable privacy and security* (pp. 13-23). ACM.

Robertson, A. (2012). *LulzSec hackers post data on 8,000 Twitter accounts, but your passwords are safe*. Retrieved January 10, 2014, from http://www.theverge.com/2012/6/12/3080534/lulzsec-reborn-twitter-tweetgif-hack

Slattery, B. (2011). *LastPass, Online Password Manager, May Have Been Hacked.* Retrieved January 10, 2014, from http://www.pcworld.com/article/227223/LastPass_Online_Password_Manager_May_Have_Been_Hacked.html

Strater, K., & Lipford, H. R. (2008, September). Strategies and struggles with privacy in an online social networking community. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1* (pp. 111-119). British Computer Society.

Stroud, D. (2008). Social networking: An age-neutral commodity—Social networking becomes a mature web application. *Journal of Direct, Data and Digital Marketing Practice*, *9*(3), 278-292.

Suki, N. M. (2012). Correlations of Perceived Flow, Perceived System Quality, Perceived Information Quality, and Perceived User Trust on Mobile Social Networking Service (SNS) Users' Loyalty. *Journal of Information Technology Research (JITR)*, *5*(2), 1-14.

Tomaiuolo, M., Poggi, A., & Franchi, E. (2013). Supporting Social Networks With Agent-Based Services. *International Journal of Virtual Communities and Social Networking (IJVCSN)*, *5*(1), 62-74.

Vijayan, J. (2007). TJX data breach: At 45.6M card numbers, it's the biggest ever. Retrieved January 10, 2014, from http://www.computerworld.com/s/article/9014782

Yan, J., Blackwell, A., Anderson, R., & Grant, A. (2004). Password memorability and security: Empirical results. *Security & Privacy, IEEE*, *2*(5), 25-31.

Yin, S. (2012). *Last.FM Joins eHarmony, LinkedIn to Celebrate Breach Week*. Retrieved January 10,

2014, from http://www.pcmag.com/article.aspx/curl/2405492

Zheleva, E., & Getoor, L. (2009, April). To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th international conference on World wide web* (pp. 531-540). ACM.

Zhou, B., Pei, J., & Luk, W. (2008). A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM SIGKDD Explorations Newsletter*, *10*(2), 12-22.