Some Privacy Enhancing Technologies for Protesting

Daniel Bosk, Guillermo Rodríguez-Cano, Benjamin Greschbach, and Sonja Buchegger

School of Computer Science and Communication,
KTH Royal Institute of Technology, Stockholm
{dbosk,gurc,bgre,buc}@kth.se

15th June 2016

Abstract

We give an overview of some possible future technologies that can be used for protests. We focus on privacy preserving tools that can be used before, under and after a protest:

- before, to organize a public protest;
- during, for the organizers and participants to communicate, within the group or to the outside world;
- after, possibly following up an event by verifying the participation and computing verifiable statistics, e.g. how many participants and in what area.

Keywords: Protesting; online social networks; privacy-enhancing technologies

Contents

1 Introduction

The rapid development of technology in the latter half of the 20th century and its increasing prevalence in everyday life has helped large parts of the world to reach the 21st century with a means of having real-time secure communications. However, the success of such development has come with some trade-offs, for example in data collection. Better storage technologies have allowed for longer data retention policies for both the private and public sectors, not only providing new and better services to combat crime, but also compromising the privacy of citizens, and sometimes their safety in oppressive regimes.

Among these technological advances, online social networks (OSNs) stand out as a popular computer-mediated tool allowing people and other entities to interact by sharing and exchanging information of any kind. Computation power and network communication are combined to make social interactions between people possible at any time and in any place lessening political, economical and geographical boundaries. Such social media are increasingly used for political activism ranging from showing one's leanings by liking something to actual support and organization of protests.

Many OSNs are run in a centralized manner — the service provider acts as a communication channel between the users of the OSN. Such structure allows providers to oversee a large portion of the data, if not all, exchanged between the users. Bearing in mind that in the case of OSNs, much of it is of personal and sensitive kind, for example, posting a picture in the network may reveal the physical geolocation as this information can be embedded in the meta-data of the image. This has proved to be problematic for political activism in several ways: a centralized ownership and control make it easier to shut down, for example, the governmental block of Twitter in Turkey for several days (Gadde 2014), or public institutions can subpoena information from the service provider. Moreover, the massive collection of data in these networks makes them an ideal target for attackers such as competitors or even governmental agencies. For example, in recent years, intelligence and security agencies of some countries have targeted these services to gain personal information about their citizens, enemies and even allies (Greenwald and MacAskill 2013). Since a centralized system can log not only data the users upload but also meta-information about their behaviour, such as online times, whom they communicate with, their location and social ties, there is a wealth of information to connect a person to a cause.

While we acknowledge the benefits of such technological advances like OSNs, we also point out the costs to personal privacy and advocate for the need to develop privacy-enhancing technologies (PETs) that can co-exist with these technologies. For example, decentralized solutions try to achieve provider independence and, in some cases, they also offer censorship resistance. Privacy-preserving solutions provide data protection by prevention, for example, by means of cryptographic techniques an organization could enforce certain policies instead of relying on the security of the system and its maintenance.

Besides providing technological support to the conventional and long-established form of protesting physically, online technologies have also opened the possibility

to alternative ways, such as virtual 'petitions', or in general, expressing support for an opinion in the form of an encouraging comment or simply affirmation.

In this chapter we focus on describing some privacy-enhancing tools in the context of OSNs that we believe can be useful in a protest and that have not yet seen a widespread use in practice. Although a protest itself relies mainly on the traditional physical act of gathering, we believe that it would benefit from some of the developments originated in the fields of information security and privacy.

1.1 The Protesting Problem

The topic of protesting is rather wide. As suggested above, it could take the form of showing support for a statement in an OSN. It could also take the form of people joining together in the streets for a demonstration. For this chapter we will consider the following scenario: Alice[1] lives in a country under the rule of an authoritarian regime. Alice wants to organize the opposition and lead a public protest to show that the people want a democratically reformed government. As expected, the authoritarian regime wants to prevent this from happening. The regime's goal is to oppress the opposition so that they cannot ever reach a big-enough protest to show the majority's dissatisfaction with the regime. They will try to stop Alice as early as possible to avoid her ideas spread throughout the population.

Although our scenario is set in an authoritarian regime, the tools and techniques described below are also useful in other systems of governance, such as democracies — because public protest and demonstrations are of importance also in democracies, to keep them democracies and not just to form them. More broadly, privacy preservation is needed also in ostensible democracies, as illustrated by whistleblowers such as Edward Snowden. A technical solution can potentially prevent privacy breaches and thus enforce legal solutions.

There are, however, some limitations to what a technical solutions can achieve. In particular we face two problems. The first one, the double agent problem, is caused by humans ability to deceive each other, and consequently cannot easily be solved by technology. The second one, the Sybil attack, is a consequence of the unlimited ability of creating arbitrary identities in online systems such as social networks.

The double agent problem is the problem of one of the regime's agents infiltrating the opposition by acting as if part of the opposition. We cannot solve this problem, however, we might be able to reduce the damage. One design principle for privacy is data minimization, this strategy will help Alice reduce the information that the double agent, and anyone else, can learn.

The Sybil attack is somewhat related to the double agent problem, but is only a problem in electronic systems. The problem occurs when there is nothing that limits the creation of new identities, thus the adversary can create multiple

[1] We use the nomenclature of the Computer Security field, where Alice tries to securely communicate with Bob and others, so-called adversaries, interfere.

unlinkable identities. Usually the attack is aimed at reputation systems, where the adversary can use its many identities to vouch for each other to falsely gain in reputation. In general, the problem can be summarized as that a rather small number of people in the network control a large part of the identities in the network in order to gain a disproportionate amount of influence (Douceur 2002).

Douceur (2002) proved that this problem cannot be solved without a *logically* central control of the the creation of identities. This means that we can not handle identity creation by letting the people who are already in the network vouch for other identities, i.e. to build a network of trust. The Sybil attack itself aims at compromising a considerable portion of the identities, and thus, the more identities the attacker gains, the more new identities it can create and vouch for. To prevent this kind of behaviour we rather need something like the national identification systems present in most countries, where the state has ensured a one-to-one correspondence between identities and physical persons. Fortunately, there are techniques that can mitigate the effects of the Sybil attack without forcing us to use such a centralized identity system. We will return to these where relevant in the text.

1.2 Outline

In this chapter we will describe some privacy-enhancing tools that we have developed and that we believe are useful in the context of protesting. We will describe them in relation to how they can be used, more specifically we categorize them as useful before, during or after a protest:

Before Organization, for example, decisions on the aim of the protest or the target audience that is expected to participate in the protest. We address some of these issues in our scenario of OSNs in Section 2.

During Communication during the protest, for example, the organizers may need to get in touch with the press over the phone during the protest. We discuss how these communications can be better protected in Section 3.

After Following up a protest by the organizers, not only to assess their success but also to correct the flaws for the next time. For example, the organizers may want to obtain reliable statistics on the number of attendees per area. We discuss different authenticity and verifiability properties of use for this stage in Section 4.

2 Before a Protest

Organizing a protest in a privacy-preserving manner does not come for free. There are many trade-offs to consider, both for Alice the organizer and the potential co-organizers and participants.

Participation We assume that Alice wants to protest in a collective manner (not alone). She will have to find interested people with whom she can co-organize the event and, later on, also to participate in the protest itself.

Finding people with similar interests can be a difficult task in general, doing so when the interests are stigmatized or not legally accepted can be much more difficult — if not impossible. For example, diversity of sexuality and gender identity are denied and even severely punished in some totalitarian regimes. If Alice wants to arrange a protest for those rights, she might be very reluctant to reveal such ideas. The plausible severe consequences for Alice to find a co-organizer, Bob, who deliberately supports and reports to the government of the regime such circumstance may oblige Alice to censor herself.

In Section 2.1 we discuss a technical solution that can make the task of finding co-organizers and potentially interested participants in a privacy-preserving manner for all parties. However, there is no technology nor solution to ensure that Bob is not lying to Alice about his interests. Therefore, we will use the term 'expressed interest' when referring to the common interest Bob has revealed to Alice.

Communication Alice and the other co-organizers will have to communicate with each other. Moreover, Alice and the co-organizers will want to spread the word about the protest to other potential participants.

A trivial solution to the communication problem is the traditional face-to-face meeting — with the trade-off that the invited attendants should be able to meet at the same time in the same place. If the requirement for synchronous communication is not that strict, then we need a way to communicate the outcome of the meeting to those who did not attend. Thus we assume that Alice will also want to communicate with Bob by means of a secure channel to avoid any non-verified third party, for example the governmental intelligence agency, to eavesdrop on her conversations. Such two-parties secure communications we discuss in Section 2.2, while in the case of more than two participants we describe in Section 2.3.

Agreement Alice and the co-organizers must agree on a time and place to hold the protest. This can also be extended to interested participants. For example, the organizers may be interested in having assurances on how many invited participants are really committed to attend the event in such a way that they do not reveal the details about the protest, such as the location, to those have not committed to attend. At the same time, the participants who have committed to attend may want to have assurances that they will be told the details of the protest if they express their commitment to the organizers. This type of property can be interesting to use in combination with a reputation system. This way we can limit the extent of the regime's possible Sybil attacks.

In Section 2.4 we discuss some aspects related to the scheduling of the event in a privacy-preserving manner.

2.1 Searching for Your Friends

So the challenge is to protect user data from malicious adversaries but at the same time making users findable for other legitimate users. To distinguish between these two cases, we assume that legitimate users possess more information about a target user than the adversary. Then a knowledge threshold can be enforced using cryptographic techniques, to guarantee that a user can only be found if the party searching for her can present enough details about her ('find me if you know enough about me').

Two protocols' implementations are presented by Greschbach, Kreitz and Buchegger (2014) that have different advantages and disadvantages. Neither rely on any central repository of user data but are suitable to be implemented in a completely decentralized way using a distributed hash table (a DHT). This avoids the biggest risk to user data: the leakage of a central database with sensitive information about a large number of people.

The proposed protocols allow users to register their identifiers (e.g. links to their profile pages, e-mail addresses or other contact information) and specify the required knowledge that is needed to find this information (e.g. name, city, workplace and date of birth). One implementation guarantees this knowledge-threshold by encoding the storage location of the registered user identifiers using the required knowledge attributes. Only users that know these attributes can construct a valid lookup request for the DHT that will return the desired user identifier. The other protocol stores user identifiers encrypted in the DHT and uses threshold secret-sharing techniques to guarantee that no user with less than the required number of attributes can decrypt a stored identifier.

Neither protocol can provide perfect protection. In the worst-case of a targeted attack, an adversary with profound background knowledge about the target user will likely succeed. For example, we cannot protect the user identifier if the adversary knows as many attributes about the target user as legitimate users do. At the same time, both schemes protect the users fairly well from large-scale crawling attacks as the search space of all possible attribute combinations is too large to brute-force and the protocols transform the registered user data in such a way that inferences from the publicly stored data are infeasible. Even if the adversary focuses her effort to only crawl the data of a specified subset of the user-base (e.g. all persons working at a specific organization), the proposed protocols offer good protection.

The knowledge-threshold is an individual user parameter, so users that consider themselves to be more exposed to risks can choose a higher knowledge-threshold to increase their protection at the cost of a lower usability, as a higher threshold makes it harder for other legitimate users to find them. In that sense, the presented protocols allow users to individually balance their findability and privacy requirements.

2.2 Communication between People

We will now focus on the communication. Specifically we will focus on communication between pairs of people, e.g. Alice talking to Bob. Borisov, Goldberg and Brewer designed a secure protocol for two-people communication, the Off-the-Record (OTR, Borisov, Goldberg and Brewer 2004) protocol. They desired an electronic equivalent of face-to-face conversations, i.e. that they leave no proofs of any kind behind: if Alice and Bob has had a conversation, Bob cannot go to Eve afterwards and prove anything about what Alice has said — the same as in a face-to-face conversation. This property is not true for email or most centralized communication services.

2.2.1 Standard Email

The standard email system does not provide any security. A suitable analogy would be that each message is a postcard, i.e. it has no envelope, so the content and address are visible on it. This means that the postman can read the cards' contents, their recipients' and senders' addresses. (Yes, unlike real postcards these also include the sender's address.) Furthermore, most postmen use transparent sacks to carry the postcards, so everyone along the way can also read the sender's and recipient's address and the contents. However, some postmen have started using non-transparent sacks, i.e. encrypted connections between the servers, so those postcards can only be read by the staff in the post-office. Thus

the email system provides no confidentiality: each email server can read the messages, each network operator along the transport route can also read (and make a copy of) each email. However, it is actually worse than that, because the email system provides no integrity either. This means that the postman, or anyone along the way, can do arbitrary modifications to the messages without anyone noticing the difference. We can safely say that we cannot rely on the email system for neither security nor privacy when planning a protest.

When using a centralized communications service, such as Facebook, the level of security and privacy we can achieve is that the postman carries non-transparent sacks. The business model of most such services is to read peoples postcards to better profile their interests and thus deliver better suiting advertising. Here, third parties cannot directly see who is communicating with whom. They can only see that something goes to and from the service. However, all information is available internally to the service. This means that there are ways of learning this, for example through PRISM (Greenwald and MacAskill 2013) of the US National Security Agency (NSA).

2.2.2 Secure Email and Text Messaging

Secure email works by employing cryptography: we encrypt the contents of the postcard, providing confidentiality, and then add a digital signature to prevent modifications. Thus the recipient is the only one who can read the message and the recipient can also verify that the message has not been modified along the way. To make key management easy, most schemes use public-key cryptography.

This means that we have two keys, one which is public and another which is kept private. For encryption, the public key can transform a message to a ciphertext, i.e. a random-looking text string. The private key can be used to transform the ciphertext back to the message. Given only the public key, it is 'impossible' to find the private key. For signatures, we can use the private key to compute a signature of a message and then send the message and its signature. The recipient can then use the public key to verify the signature of the message. This signature depends on the entire message, so it is impossible to move a signature to another message — unlike signatures on paper. And since it is impossible to find the private key given only the public key, no one can create fake signatures.

One problem with this approach to secure email is that the sender and recipient are still in the clear, anyone can read them. So the content is hidden, but the meta-data is not.

Another problem is that the digital signatures used provides a property called non-repudiation. Say that Alice securely sent an email to Bob, if Eve would compromise Bob's private key, as many government agencies can, then she would learn that Alice — and no one else — has sent that message to Bob. Bob might even give the message and his key to Eve voluntarily or under threat. This is exactly the property that Borisov, Goldberg and Brewer wanted to remove with OTR. They can do this by leveraging the interactive nature of instant messaging (IM) and changing the digital signatures to shared-key message-authentication codes (MACs). Shared-key means that Alice and Bob share the same key for generating and verifying a MAC. This means that Bob can generate valid MACs for any message and show to Eve, thus he cannot prove to Eve what Alice has said — since he could have created this 'proof' himself. In addition, Alice and Bob do not use the same MAC key throughout their conversation, then continuously exchange new keys, one for each message. However, in this situation, Eve still has only two candidates as the author of the message: Alice and Bob, since they both have access to the shared keys. To remedy this problem Alice and Bob publishes the MAC keys after use, i.e. when they no longer need them. This gives 'everyone' the possibility of generating messages that verifies under Alice and Bob's key, so now Alice and Bob can argue that someone (Eve included) could have modified the ciphertext.

The OTR protocol became widely spread after the 2013 revelations about the mass surveillance of the NSA and UK Government Communications Headquarters (GCHQ), many derivatives of the protocol emerged in smartphone apps. Among the most wide-spread derivatives of OTR is Signal (formerly TextSecure) (*Signal Private Messenger*)[2]. The Signal protocol has, unlike many other of the derivatives, been formally analysed and proven that it indeed provides its claimed security properties (Frosch et al. 2016). One improvement over OTR is the deniability. In Signal the authentication is set up in such a way that any person knowing the public key of Alice and Bob can generate a fake transcript

[2]TextSecure actually existed before the Snowden revelations, but has seen more wide-spread use after.

of a conversation. This results in that Eve has many more candidates for the authors of a conversation.

2.2.3 When the Adversary Controls the Network

Bosk, Kjellqvist and Buchegger (2015) argue that if the adversary controls the entire network, then the approach to deniability taken by OTR and Signal does not suffice. The problem is that the adversary can record a transcript of all communications that have taken place. We know that the NSA did exactly that (Greenwald 2013), and specifically saved ciphertexts for later when the decryption key might be available. In this setting it does not matter if anyone can generate a false transcript of a conversation between Alice and Bob, the regime knows exactly what Alice has sent and Bob received and vice versa. The argument of OTR-like schemes is that Alice and Bob has the possibility to deny anything about the conversation since it cannot be decrypted.

There are more than one way to approach this problem. The first approach would be to use an anonymizing service, such as Tor (Dingledine, Mathewson and Syverson 2004). This way, the regime would not know that Alice communicates with Bob, only that Alice communicates with someone. However, for all low-latency solutions, when the entry point and exit from the anonymizing network are both controlled by the adversary, then the adversary can perform a correlation attack and essentially render the anonymization service useless (Danezis, Diaz and Syverson 2010). This is in fact the case if the regime controls the nation-wide network while critics of the regime, all located in the country, want to communicate in real-time. To make this attack more difficult for the regime's surveillance agency we must introduce random delays in our communication. And despite all this, the regime can still ask Alice to decrypt the conversations — either she complies or claims she do not know the key.

The second approach would be to ensure deniability even against this strong adversary. This would not hide who communicates with whom, as in our first approach, but it provides deniability for the conversations. The scheme suggested by Bosk, Kjellqvist and Buchegger (2015) makes use of one practical instance of deniable encryption (Canetti et al. 1997). They construct a scheme where Alice and Bob can create 'false witnesses' for their conversation. Basically Alice can create a decryption key such that when used to decrypt the ciphertext recorded by the regime from the network it will decrypt to a plaintext of Alice's choice. This way she can 'prove' her innocense. However, the question whether the regime would actually accept such a 'proof', knowing it can equally well be false, remains open.

2.3 Holding Discussions

So far we have treated only two-party conversations, i.e. Alice and Bob talking to each other. However, there are usually more than two people organizing a protest, and so we need to hold discussions with more than only two people at a time. In this situation there are two approaches to solving the communic-

ation: simultaneous pair-wise communication between all participants or true group communication. Furthermore, how the messages are distributed is also important, because the adversary can learn who the participants are.

2.3.1 Group Communication Properties

When a group uses pair-wise communication, every member of the group will set up a pair-wise channel to each other member of the group. Each pair-wise channel is as described above, in Section 2.2. Then for every message Alice wants to send to the group she has to send it to every participant. This would allow Alice to cheat, e.g. she can send 'Who wants to overthrow the regime?' to everyone except to Bob, to whom she instead sends 'Who wants to order pizza?'. This opens up for the Byzantine Generals' problem (Lamport, Shostak and Pease 1982), where malicious actors can lie to honest actors to disrupt operation. Lamport, Shostak and Pease (1982) in fact proved that it is impossible for the honest parties to recover and identify the malicious parties if the malicious parties exceed a third of the participants.

Although Alice's ability to say different things to different participants is in itself a desirable property from Alice's perspective — she would like to lie to suspected regime agents — this property can at the same time be undesirable due to the Byzantine Generals' problem. For this reason group communication must provide better properties, namely that everyone hears who said what and when, thus forcing Alice to say the same thing to all participants. In such a scheme, when Bob replies 'I do, shall we say tonight?' the others will see that Bob is replying to something they did not see and not to the question 'Who wants to overthrow the regime?'.

Goldberg et al. (2009) tried to extend the OTR protocol to a multi-party setting. This did not result in a concrete protocol implementation, and the resulting protocol they suggested was also very complex. It also had some undesirable limitations, for instance, the scenario that Bob receives a question which is different from everyone else's is only detected at the very end of the conversation. As is pointed out by Marlinspike (2014), asynchronous communication today has no real end, which makes the approach of Goldberg et al. (2009) even less appealing. Due to this, Open Whisper Systems (*Signal Private Messenger*) implements group chats as simple pair-wise conversations. With additional meta-data they can ensure consistent history, however, this is not yet implemented (Marlinspike 2014). A technique that could be used for this is to include a message digest of the entire conversation history with each message. A message digest is computed using a one-way function, i.e. its output is unpredictable and its input is impossible to compute given only the output. This means that the message digest included in Bob's reply and the one computed by the other participants above would differ, thus everyone learns that the conversation history is inconsistent and should no longer be trusted. Due to the unpredictable property of the one-way function, Alice cannot phrase the two different messages in such a way that they yield the same message digest in the history either. But despite this, the other participants cannot determine if it

is Alice or Bob who is lying about the message history — Alice could send the same message to everyone and still Bob could try to frame her.

2.3.2 Message Distribution

Bosk and Buchegger (2016) analysed two dichotomous models of communication. The first was the pull model, where the recipients fetch (i.e. pull) new messages from the sender. A suitable analogy would be that of magazines published through sales in kiosks: people go to the kiosk to get the latest publications. The second model was the push model, here a sender sends the message directly to the recipients. Thus it is more like magazine subscriptions: the next issue arrives in the mailbox shortly after publication. This is the model of the communication described in Section 2.3.1, i.e. the communication model for email. Bosk and Buchegger found that achieving privacy in the pull model is technically easier than in the push model. In fact, achieving strong privacy in the push model is very difficult.

The Pull Model We can start by looking at the pull model for communication. Alice wants to distribute a message to the participants in a discussion. In the pull model the participants actively ask Alice or an intermediary for new messages at regular intervals. To form a protocol around this model, Alice and the participants can agree on a location for the messages. When Alice wants to send a new message, she writes it to this particular location. When the other participants want to, they can read from the location to see if there are any new messages.

We can assume that the network that Alice uses is controlled by the regime's agents. We can also assume that the storage where the messages are stored is publicly readable, because it is probable that the regime's agents can compromise it. Thus Alice does not want to be associated with the message, not authorship nor posting it.

The first thing we can say about this is that Alice would like to have confidentiality for the message contents, so the regime's agents cannot read it. She would also like to have integrity for the message, so that the recipients can be sure that the regime's agents have not modified it. Many systems provide these two properties, e.g. Pretty Good Privacy (PGP) which is used for email. However, Alice also wants to hide the sender and recipients, a property which PGP and others do not provide. There is a class of encryption schemes called anonymous broadcast encryption (ANOBE, Libert, Paterson and Quaglia 2012) schemes. This type of scheme provides confidentiality while hiding the sender and the intended recipients. If Alice can write the message anonymously to the storage and the message is encrypted using an ANOBE scheme, then it will be difficult to determine the sender. Furthermore, if the recipients fetch the messages anonymously too, then the recipients are also hidden. Now the problem of integrity remains. If Alice and the other participants agree on a commonly shared message authentication (MA) key, then they can use MACs to ensure integrity. The reason MACs are more desirable than digital signatures in this

situation is that anyone who can verify the authenticity of a MAC can also create one (as was pointed out above). With digital signatures on the other hand, if Alice signs a message it is clear that Alice is the only one who could have signed it. This would provide the regime's agents with something to track messages by, all messages signed by the same key are related. With MACs, any of the other participants could also be the author of the message and the regime cannot determine which messages are related either. This means that for a discussion, any of the participants would be equally likely to be the author of a given message. However, this relies on the anonymity of the actors.

The Push Model In the push model Alice will send the message to all recipients directly. The problems with the push model is that it reveals more meta-information. Say that all participants have an inbox, similarly as in the email system. Assume that the regime's agents monitor the network on a national level. In this situation they can see one message originating from Alice, going to a server beyond the agency's reach, and soon n equally-sized messages return from the server near-simultaneously addressed to n people. This is what is called a time-correlation attack. The agency can then relate Alice with the people she is talking to. This is the situation when using text-messaging apps such as Signal, because there all messages go through a central server. This also has the side-effect that the operator of the central server can perform the same attack.

Now let us try to remove some information to make this more difficult for the agency. Say that the sender and recipients are anonymous, so the agency can only track which messages ends up in which inboxes and when. Despite the anonymity they can still do a time-correlation attack on the incoming messages. This means that they learn which messages are related and in turn which inboxes are related and how. This means that the agency have the structure, they only need to fill in the identities. If one of the participants makes a mistake, then the regime's agents will have a starting point to target. For example if a participant uses the same inbox for communication with all his friends, and not only the participants in the plot against the regime, then one of his other contacts might not be as concerned with staying anonymous. The consequence is that the regime can see the identity of someone sending messages to an inbox of their interest. Then they can target this person and learn which friend owns the inbox of interest. Then they can proceed to targeting one of the protest organizers. This type of attack will not work when the communication is according to the pull model, since there the agency must attack each anonymous connection.

2.4 Scheduling a Protest

There are some tasks that the organizers must accomplish prior to the protest itself. For example, they must decide who are the most suitable candidates to attend the event, how to let them know about the protest and what preliminary information they should get. They must also decide whether invitees should

learn about the attendance of other invitees or not. Preferably, all this should be possible in a privacy-preserving fashion.

Realizing this standard feature of OSNs in a decentralized manner is not trivial, because there is no trusted third party which both organizers and invitees can rely on. Since they a[illegible]epend on only themselves, an implementation of this feature must provide security properties that guarantee fairness to all parties involved, e.g. a protester can verify that the invitation she received was actually sent by the organizers. Moreover, the implementation should also provide privacy settings to protect personal information such as the identities of the participants, e.g. the organizers can restrict to only the invited participants to learn how many others have been invited, and only after a protester has agreed and committed formally to attend the event, she can learn the identities of other invited protesters.

The challenge of implementing this feature without a trusted third party becomes greater when the organizers want to allow different types of information about the event to be shared with different groups of protesters in a secure way because any participant should be able to verify the results to detect any possible cheating. For example, a neutral trusted broker, such as the organizers, could keep certain information secret, such as the identities of the invited protesters, and only disclose it to those ones who commit to attend the protest.

In the scheme by Rodrıguez-Cano, Greschbach and Buchegger (2014), they describe and formalize the security and privacy properties outlined above. More specifically, that the organizer is able to configure who can learn the identities of the invited or attending participants, or a more restrictive version where only the number of invitees or attendees are revealed. There is also an attendee-only property that guarantees exclusive access to some data, e.g. the location of the protest. These properties are accomplished using several simple primitives: storage location indirection, controlled ciphertext inference and a commit-disclose protocol. Storage location indirection allows the organizer to control not only who can read some the contents of some encrypted data, but also who can access the ciphertext. Controlled ciphertext inference can be used by the organizer to allow for controlled information leaks. This is needed to achieve properties such as revealing the number of invitees but keeping their identities secret. Finally, the commit-disclose protocol can make some secret available for only those participants who committed to attend the protest while, at the same time, detect any misbehaving party.

3 During a Protest

There are a few aspects that we must cover which relate to this part of our treatment. During a protest, organizers and demonstrators might want to communicate, either among themselves or to the outside world. The communication to the outside world can have at least two purposes. The first one is simply to try to get more people to come to the demonstration. The second is when a demonstrator wants to store something for posterity. This can be a photo cap-

turing police brutality or a part of a proof-of-demonstration (as in Section 4).

There are a few problems with communication during a protest. If the participants use the phone network, i.e. infrastructure which is generally controlled by the government, participants can be tracked and bound to the location by their phone. If they communicate over the phone network they can still use the techniques outlined in Section 2. However, if they do not want to be tracked, there are two options:

1. they must use another network infrastructure that is not controlled by the government,
2. the mechanisms in Sections 2 and 4 must allow executions without access to a global communications infrastructure during the demonstration.

There are solutions to the first problem: wireless ad-hoc networks. The area of ad-hoc networks is far too wide for us to convey more than the general idea of the field in this chapter. The idea of ad-hoc networks is to form a network using ad-hoc connections. For example, if Alice can communicate with Bob, Bob in turn can communicate with both Alice and Carol, then Alice can communicate with Carol through Bob. Protesters can use this technique to form an ad-hoc network at the physical location of the demonstration, thus avoiding the government controlled network. Depending on the reach of the ad-hoc network, participants might get access to the global Internet through some node in the network. If not, they are limited to communicating only between themselves.

For the second problem listed above, we can achieve local communication, e.g. pair-wise communication through bluetooth or Near-Field Communication (NFC). So we can assume that this possibility exists. However, communicating certain data to the outside world will still be the ultimate goal, but we will use the local communication possibilities to ensure the authenticity of the data after the demonstration is over. We will discuss this further in Section 4.

4 After a Protest

The main goal after a protest is to provide verifiable data. For instance, how can we ensure that photos from a demonstration are authentic? We can probably recognize the place the photo is portraying, however, the meta-data such as time-stamps of the file can be manipulated. So the only thing we can say is that the photo was taken at the latest at the time of publication. Similarly, how can we determine the number of participants of a demonstration? Many techniques used today estimate the number given photos of the demonstration, but this might not give an accurate number. And, as mentioned, the source of the photo must be trusted, otherwise the authenticity of the photo can be questioned.

We will discuss these two problems in this section. First we will discuss the problem of data authenticity in Section 4.1, i.e. that the data can be correctly tied to the demonstration. Then we will discuss verification of the participation of a protest in Section 4.2, i.e. the ability to compute and verify the number of participants.

4.1 Data Authenticity

The problem of authentically associating data with a physical event is difficult. We can essentially divide it into the following requirements:

1. Prove that the data was created before the end of the event.
2. Prove that the data was created after the start of the event.
3. Prove that the data is spatially related to the physical location of the event.

Requirements 1 and 2 together bind the data to the time of the event whereas requirement 3 binds the data spatially to the event.

Now, consider the scenario of Alice taking a photo during the demonstration and posting it online. What can we say about this photo? First, we can say that it was created before we viewed it, so requirement 1 is fulfilled if we view it in relation to the event. If it was submitted to a service that we trust, then we can also trust the time-stamp of the service. Furthermore, we can also consider it spatially related to the physical location (requirement 3) if we can convince ourselves that the photo is depicting the physical location and not any kind of 'reconstruction', e.g. it is computer generated or a photo of a similarly looking location.

Requirement 2 is more difficult to achieve. In the above scenario, there is nothing preventing the photo from being much older than the event yet spatially related to the physical location. In the security field requirement 2 is captured by a property called *freshness*. The freshness property is usually achieved by requiring that the data depend on an unpredictable value. The unpredictable value is commonly a value chosen randomly by the verifier, but the main basic requirement is that it is not under the prover's control, i.e. Alice in this scenario. For example, we might require Alice to include the front-page of a particular newspaper in the photo, since the exact front-page is difficult for Alice to predict in advance. There is a subfield of digital forensics that work with image manipulation detection, so there are methods that would prevent at least Alice's easiest manipulation attempts.

We can see that the techniques to achieve requirements 1 to 3 depends on the type of data. We used as an example above a photo, next we will look at another type of data.

4.2 Verification of Protest Participation

One problem in physical protesting that has not yet been solved satisfactorily is the crowd counting problem, or more generally, the verification of the participation in a protest. After many protests the demonstrator count by police and that by organizers differ, in some instances the difference can be hundreds of thousands. Many of the methods to count the participants in the literature are based on computer vision, i.e. object recognition through image analysis. With this type of technique a third party has no way of verifiying that the count is

correct, hence the dispute between organizers and police over these counts. A demonstration is very similar to voting, both are many individuals expressing their opinion. Hence it is desirable to have similar properties for verifying the participation in a protest, where this verification step is at the core.

In the context of voting protocols, in particular e-voting protocols, there are three desirable properties for verification:

1. Anyone can verify that the result is according to the cast votes.
2. Anyone can verify that each vote cast is legitimate.
3. Every voter can verify that its vote is included in the result.

We can transfer these properties to the case of protest participation, then each vote would be replaced by a proof of participation. Property 2 would in this case mean that anyone can verify that each participation proof belongs to a unique individual, i.e. to prevent any Sybil attack (Section 1.1).

The three verifiability properties above are indeed desirable, e.g. then the United Nations (UN) can verify protests happening in a country and the country cannot deny it, thus the UN can apply pressure if needed. However, the properties are difficult to accomplish with computer vision methods, especially if nobody from the UN has been on location to collect the image material for the analysis — since then the authenticity can be questioned as outlined in Section 4.1.

Additionally, in voting, the cast votes are not linkable to the identity of the casters. Thus it is not a problem to reveal the identites of those who participated in the vote, since they could have voted for any alternative. We would also like to have the corresponding property when verifying the protest. The very nature of a protest is different though, we do not have any choice: if we participate we support the cause of the protest. Consequently we want to verify the participation of a protest without identifying individuals who participated. Otherwise the regime's agents can identity all the participants and simply 'make them disappear'.

Finally, for real-world protests we need to bind participants to the same physical location at a reasonably similar time, i.e. within the area and duration of the protest. This means that for a system like this to work, we also need the requirements from Section 4.1: proof that the data was created after the start of the protest, proof that it was created before the end, and proof of spatial relation to the location. For the last property, Gambs et al. (2014) developed a decentralized location-proof share (LPS) which provides a participant with a verifiable proof of having been at a location at a certain time, something we call a location proof (an LP). It is decentralized because there is no central authority that vouches for the location, instead peers act as witnesses. Then a third-party can verify the authenticity of the LP, by verifying the witnesses signatures, and can thus be sure that the person has indeed been in the location. Bosk, Gambs and Buchegger are currently exploring (work in progress) the possibility of combining such an LPS with the verifiability properties for voting protocols

into a system for verifying the participation in protests. The overall idea is that each participant generates an LP during the protest, where (some of) the other protesters act as witnesses, then the LPs can be used to compute the participation count with all the verifiability properties above.

5 Conclusions

In this paper we presented some privacy-enhancing technologies that can be relevant for political activism, with a focus on supporting various kinds of protests from online organization to evaluation after the fact. While we discussed a range of technologies, this selection presents only a small part of what has been conceived of in terms of privacy and transparency enhancing technologies in the greater research community over the last decades.

There still is a large gap to bridge between privacy researchers and political activists. As in the general population, not many of the existing technologies for privacy are actually used, and those that are used (Tor being such a notable exception) are not often used widely. There are several reasons for this, in our opinion. The main reason is perhaps a lack of communication. It is difficult to communicate across different academic disciplines and between academia and the outside world. People may not have heard of relevant technologies, may be reluctant to use them, or may not see their use case reflected. The so-called engineer's disease of offering technical solutions to social problems is reflected in the quite common lack of thorough analysis of what actual users need. Usability is another issue that will need to be taken into account more, requiring interdisciplinary work, communication, and user participation. Many of the existing technologies, while conceptually excellent, only exist as prototypes, or as isolated primitives that are not integrated into a usable and comprehensive service.

There is much work left to connect the existing technologies with actual people that can benefit from them, and vice versa, researchers lack input on what is actually needed in the field and often scalable ways of turning academic prototypes into complete and secure tools to protect the privacy of the people.

References

Borisov, Nikita, Ian Goldberg and Eric Brewer (2004). 'Off-the-record communication, or, why not to use PGP'. In: *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*. ACM, pp. 77–84.

Bosk, Daniel and Sonja Buchegger (2016). 'Privacy-Preserving Access Control for Publicly Readable Storage Systems'. In: *IFIP Summer School 2015*. Ed. by David Aspinall, Marit Hansen and Jan Camenisch. Lecture Notes in Computer Science. To appear. Springer.

Bosk, Daniel, Martin Kjellqvist and Sonja Buchegger (2015). 'Towards Perfectly Secure and Deniable Communication Using an NFC-Based Key-Exchange Scheme'. In: *Secure IT Systems*. Ed. by Sonja Buchegger and Mads Dam. Vol. 9417. Lecture Notes in Computer Science. Springer International Publishing, pp. 72–87. ISBN: 978-3-319-26501-8. DOI: 10.1007/978-3-319-26502-5_6. URL: http://dx.doi.org/10.1007/978-3-319-26502-5_6.

Canetti, Ran, Cynthia Dwork, Moni Naor and Rafail Ostrovsky (1997). 'Deniable encryption'. In: *Advances in Cryptology—CRYPTO'97*. Springer, pp. 90–104.

Danezis, George, Claudia Diaz and Paul F. Syverson (2010). 'Systems for Anonymous Communication'. In: *CRC Handbook of Financial Cryptography and Security*. Ed. by B. Rosenberg and D. Stinson. CRC Cryptography and Network Security Series. Chapman & Hall, pp. 341–390.

Dingledine, Roger, Nick Mathewson and Paul F. Syverson (2004). 'Tor: The Second-Generation Onion Router'. In: *USENIX Security Symposium*, pp. 303–320.

Douceur, John R. (2002). 'The Sybil Attack'. English. In: *Peer-to-Peer Systems*. Ed. by Peter Druschel, Frans Kaashoek and Antony Rowstron. Vol. 2429. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 251–260. ISBN: 978-3-540-44179-3. DOI: 10.1007/3-540-45748-8_24. URL: http://dx.doi.org/10.1007/3-540-45748-8_24.

Frosch, Tilman, Christian Mainka, Christoph Bader, Florian Bergsma, Joerg Schwenk and Thorsten Holz (2016). 'How Secure is TextSecure?' In: *European Symposium on Security and Privacy*. Full version as IACR ePrint 2014/904. IEEE. URL: http://eprint.iacr.org/2014/904.

Gadde, Vijaya (2014). 'Challenging the access ban in Turkey'. In: *Twitter Inc.* Fetched on 14th June 2016. URL: https://blog.twitter.com/2014/challenging-the-access-ban-in-turkey.

Gambs, S., M.-O. Killijian, M. Roy and M. Traore (2014). 'PROPS: A PRivacy-preserving lOcation Proof System'. In: *Reliable Distributed Systems (SRDS), 2014 IEEE 33rd International Symposium on*, pp. 1–10. DOI: 10.1109/SRDS.2014.37. URL: http://ieeexplore.ieee.org/ielx7/6979347/6983362/06983374.pdf.

Goldberg, Ian, Berkant Ustaoğlu, Matthew D Van Gundy and Hao Chen (2009). 'Multi-party off-the-record messaging'. In: *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, pp. 358–368.

Greenwald, Glenn (2013). 'XKeyscore: NSA tool collects 'nearly everything a user does on the internet''. In: *The Guardian*. URL: http://www.theguardian.com/world/2013/jul/31/nsa-top-secret-program-online-data.

Greenwald, Glenn and Ewen MacAskill (2013). 'NSA Prism program taps in to user data of Apple, Google and others'. In: *The Guardian*. URL: http://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data.

Greschbach, Benjamin, Gunnar Kreitz and Sonja Buchegger (2014). 'User Search with Knowledge Thresholds in Decentralized Online Social Networks'. English. In: *Privacy and Identity Management for Emerging Services and Tech-*

nologies. Ed. by Marit Hansen, Jaap-Henk Hoepman, Ronald Leenes and Diane Whitehouse. Vol. 421. IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, pp. 188–202. ISBN: 978-3-642-55136-9. DOI: 10.1007/978-3-642-55137-6_15. URL: http://dx.doi.org/10.1007/978-3-642-55137-6_15.

Lamport, Leslie, Robert Shostak and Marshall Pease (1982). 'The Byzantine Generals Problem'. In: *ACM Trans. Program. Lang. Syst.* 4.3, pp. 382–401. ISSN: 0164-0925. DOI: 10.1145/357172.357176. URL: http://doi.acm.org/10.1145/357172.357176.

Libert, Benoît, Kenneth G Paterson and Elizabeth A Quaglia (2012). 'Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model'. In: *Public Key Cryptography–PKC 2012*. Springer, pp. 206–224. URL: http://eprint.iacr.org/2011/476.

Marlinspike, Moxie (2014). *Private Group Messaging*. URL: https://whispersystems.org/blog/private-groups/.

Open Whisper Systems. *Signal Private Messenger*. Accessed on 18th April 2016. URL: https://play.google.com/store/apps/details?id=org.thoughtcrime.securesms&hl=en.

Rodrıguez-Cano, Guillermo, Benjamin Greschbach and Sonja Buchegger (2014). 'Event Invitations in Privacy-Preserving DOSNs: Formalization and Protocol Design'. In: *Secure IT Systems, 19th Nordic Conference, NordSec 2014*. Ed. by Karin Bernsmed and Simone Fischer-Hübner. Springer, pp. 291–292. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-153741.

A Biographies

Daniel Bosk is a doctoral student at KTH Royal Institute of Technology, Stockholm, Sweden, and a lecturer of Computer Engineering at Mid Sweden University, Sundsvall, Sweden. He holds a Master of Science in Computer Science from KTH Royal Institute of Technology and a Master of Education in Mathematics and Computer Science from Stockholm University, Sweden. His research interests are in security and privacy in decentralized systems, and specifically the empowerment of the users.

Guillermo Rodríguez-Cano is a doctoral student at KTH Royal Institute of Technology. He holds a Master of Science in Computer Science from Uppsala University, Sweden, and a Bachelor of Engineering in Computer Science from University of Valladolid, Spain. His research interests lie in the area of privacy and security in social systems and modelling, data mining, and information propagation in social networks.

Benjamin Greschbach ...

Sonja Buchegger is an associate professor of Computer Science at KTH Royal Institute of Technology. Prior to KTH, she was a senior research scientist at Deutsche Telekom Laboratories in Berlin, a post-doctoral scholar at the University of California at Berkeley, and a researcher at the IBM Zurich Research Laboratory. Her Ph.D. is in Communication Systems from EPFL, Lausanne,

Switzerland, and she graduated in Computer Science from the University of Klagenfurt, Austria. Her main research interests are in privacy and decentralized systems.