A privacy-preserving e-participation framework allowing citizen opinion analysis

Francesco Buccafurri*, Lidia Fotia and Gianluca Lax

DIIES,
University “Mediterranea” of Reggio Calabria,
Via Graziella, Località Feo di Vito,
89122 Reggio Calabria, Italy
E-mail: bucca@unirc.it
E-mail: lidia.fotia@unirc.it
E-mail: lax@unirc.it
*Corresponding author

Abstract: In e-democracy, e-participation represents a key component, as it is the way to adapt government decisions to the real expectations of citizens. The availability of information-communication technologies represents the basis for the implementation of concrete plans of citizens’ participation to the government of the community. However, there is a non-trivial trade-off to manage, between security and privacy needs and opinion analysis opportunities. Indeed, whereas we have to guarantee that the action of citizens is kept anonymous, relating opinions to information about people allows the government management to coherently orient the executive action. In this paper, we present a solution of the above trade-off, by proposing a framework relying on existing social networks and working through cryptographic protocols able to ensure citizens’ anonymity yet enabling opinion analysis. A careful security analysis and the addressing of the main implementation issues make the proposal ready to a secure and feasible adoption in real-life contexts.

Keywords: privacy; e-democracy; social network; e-participation.


Biographical notes: Francesco Buccafurri is a Full Professor of Computer Science at the University Mediterranea of Reggio Calabria, Italy. In 1995 he took the PhD in Computer Science at the University of Calabria. His research interests include deductive-databases, knowledge-representation and non-monotonic reasoning, model checking, information security, data compression, data streams, agents, P2P systems. He has published several papers in top-level international journals and conference proceedings. He is Associate Editor of Information Sciences (Elsevier), is also included in the editorial board of a number of international journals and played the role of PC chair in some international conferences.

Lidia Fotia is a post-doc fellow at the University Mediterranea of Reggio Calabria. She received her MSc in Telecommunication Engineering from the University
1 Introduction

The availability of information-communication technologies, new media, devices, bandwidth, computer literacy, represents the basis for the implementation of concrete plans of citizens’ participation to the government of the community (Sharma, 2004; Rose and Sæbø, 2008; Persson and Goldkuhl, 2010). In e-democracy, e-participation represents a key component, as it is the way to adapt government decisions to the real expectations of citizens (As-Saber et al., 2007; Medaglia, 2012; Susha and Grönlund, 2012). Thus, we expect that the almost continuous presence of people on social networks possibly through smart phones and tablet will be soon a formidable chance for government entities to frequently collect opinions, preferences, evaluations, also considering that the demand of participation of citizens to the government is dramatically increasing.

This way, our communities can really evolve towards the new model of smart city, recalling that this concept has to be intended in an extended way, thus not necessarily limiting its scope just to a city, but to an entire community which could be sometime really a city, sometimes a region, sometimes an entire country.

In this process, there is a important trade-off to manage. On the one hand, we have to guarantee that the action of citizens is kept anonymous, at least to be sure that it is actually free from conditioning. In general, all the basic properties of e-voting systems (Burmester and Magkos, 2003; Pieprzyk et al., 2003), namely secreteness, uniqueness, verifiability, uncloneability, robustness and scalability, are essential requirements, as citizens’ opinions can be view as a form of vote. On the other hand, we should not miss all the powerful knowledge that we can obtain by relating opinions with information about people, allowing the government management to analyse the communities and to coherently orient the executive action. To have the concrete perception of the above trade-off, consider some real-life examples of citizen participation, like the preliminary evaluation of a law or a reform, a political parties poll, a satisfaction survey, a primary election, just to mention a few. In all these cases, enabling the possibility of analysing the results by correlating
them with information about citizens, like job, age, religion, salary, region, etc. gives very strategic information, whose importance is comparable to that of the opinions themselves. Consider for instance the opinions expressed in case of a preliminary evaluation of reform. The analysis of the social-economic composition of the survey result can be used to drive the modification of the reform in a more effective way than taking into account only the survey result itself.

The solution of the above trade-off between security and privacy needs and opinion analysis opportunities is certainly not trivial. In principle, concerning the basic properties of e-voting systems, we could apply one of the solutions existing in the literature. However, the features provided by e-voting systems (Burmester and Magkos, 2003; Pieprzyk et al., 2003) are excessive w.r.t. the application we are considering, since some of these features have a high cost in terms of complexity of a real system implementation. Therefore, we have to find some lighter method, guaranteeing the above basic security features, whose implementation does not require dedicated infrastructures and complex working, making the solution unproportionate w.r.t. the problem. To do this, a solution based on pre-existent social networks, allowing citizens to vote through their own profile could be adopted, as done by Buccafurri et al. (2012). This way, no complex overhead is required besides an electronic card to identify a citizen or any identity management system able to identify people (plausibly, we can consider this is for free in an e-government context), and owing a profile by each voter in one of the existing social networks. The solution, among the features typically satisfied in e-voting systems, guarantees uniqueness, secreteness, verifiability, uncloneability, robustness and scalability.

Unfortunately, the lightweight e-voting approach referred above, being designed according to standard requirements, cannot address the addition problem of preserving opinion analysis. Indeed, secreteness inhibits the possibility to relate the preferences expressed by citizens even to non-identifying attributes.

In this paper, starting from a preliminary idea presented by Buccafurri et al. (2013), we overcome the above limitation by re-interpreting the classical concept of secreteness in such a way that an opinion expressed by a citizen is related to a number of attributes chosen by the citizen whose identity cannot be discovered. The attributes are certified by a Third Trusted Party, to avoid that fake information may invalidate the significance of analysis activities. Besides the possibility of analysing citizens’ preferences and extracting useful knowledge from them, our solution enables also filtering mechanisms aimed at collecting only preferences of a certain segment of the population, like all people with a certain age range, a certain job, a given region and so on. Observe that the above requirements evokes what is provided by selective disclosure and bit commitment approaches (Brands, 2000; Persiano and Visconti, 2000; Jarvis, 2003; Zwierko and Kotulski, 2007), but a direct application of such approaches to our case is not resolutive since the secret used by a citizen to enable the disclosure of the chosen attribute would allow third parties to trace the citizen herself, thus breaking anonymity. The solution presented in this paper is based on an original protocol relying on cryptographic notions like the robustness of the discrete logarithm problem and the mechanism of digital signatures and partially blind signatures. Besides a careful formalisation of the protocol, this paper includes a detailed security analysis to show that all the requirement are satisfied and addresses a number of key implementation issues, to provide a concrete contribution than a theoretical one.

The paper is organised as follows. In the next section, we contextualise our paper in the scientific literature. Section 2 is devoted to the related literature. In Section 3, we give a brief overview of our proposal. The protocol allowing the selective disclosure of some
attribute in an e-voting session is defined in Section 4. In Section 5 we illustrate all data regarding the e-voting session and the pseudocode of the protocol. In Section 6, the analysis of the security of this protocol is presented, by showing that all the desired features are guaranteed also against possible attacks. Finally, in Section 7, we draw our conclusions and sketch possible future work.

2 Related work

In this section, we survey the literature related to the topics of e-participation and e-voting, which our proposal is clearly related to, and focus on selective disclosure which represents a key issue in our approach.

According to the definition given by Macintosh (2004), e-participation refers to the use of information and communication technologies (ICT) to support democratic decision-making. It is related to the issues of enabling opportunities for consultation and dialogue between government and citizens by using a range of ICT tools (Epstein et al., 2014).

The emergence of new forms of citizen participation in political activity through ICT has attracted attention from both research and practitioner communities, as shown by the increasing of government-initiated e-participation projects at all levels and of research contributions populating the scientific literature. In the past half decade, after an initial phase characterised by experimentation, initiatives of e-participation have started to consolidate. In recent years, an increase in the degree of achievement of sustainable combinations between legacy and emerging technologies for citizen participation has arisen, as proposed in several e-participation projects (Bicking et al., 2011; Koussouris et al., 2011).

The growing body of knowledge on e-participation has also confirmed the interdisciplinarity of this field. Contributions come from the areas of both social sciences and information systems research. Perspectives from political science, sociology, management, psychology, and economics stand beside contributions that are more technical in nature. Such a scenario of disciplinary backgrounds is also accompanied by a variety of methodological stances and normative perspectives characterising e-participation research (Al-Mamari et al., 2013; Parvez, 2006).

Several attempts to implement e-participation have been carried out. Quental and Gouveia (2014) propose a Web-application designed to monitor and discuss the government activity and new laws at national, regional or local level. Since several local governments use Websites as a public relations tool, Royo et al. (2014) analyse the Websites of the environment departments of European local governments that signed the Aalborg Commitments to determine the extent to which they are using the Internet to promote e-participation in environmental topics and to identify the drivers of these developments. An empirical assessment of how online networking affects two economically relevant aspects of social capital (i.e., trust and sociability) is conducted by Sabatini and Sarracino (2014). The result of this study is that participation in social networks such as Facebook and Twitter has a positive effect on face-to-face interactions. Based on a literature review of social capital and citizen participation, Lee and Kim (2014) develop a model of active e-participation and argue that three dimensions of social capital and citizen participation management are positively associated with active e-participation. By an experimental evaluation, this study finds that active e-participation is positively affected by citizens’ trust in government, their volunteer experiences, weak off-line social ties, and perceived quality responsiveness during the e-participation process. A system, called ABC4Trust, which allows the participation of
A privacy-preserving e-participation framework allowing citizen certified students in remote course evaluations has been proposed by Liagkou et al. (2014). This system has been tested at the University of Patras (Greece) in two courses of computer science engineering. This proposal is very close to our approach. The main differences are that ABC4Trust is not general purposed as our protocol, as it can be adopted only in limited environments. Indeed, a network of contactless smart card reader is required and a smart card has to be used by each student to store credentials attesting his/her appearance in class. In our approach, to reduce the cost of the implementation, the right to vote is obtained collaboratively from a number of credential issuer users. Another significative difference with our proposal is that ABC4Trust does not include the feature of our approach giving flexibility to the end-user in the choice of the non-identifying disclosed attributes. From this point of view, ABC4Trust is much less oriented than our protocol to voting analysis, whose importance in a full e-government solution is unquestionable.

Our paper is contextualised also in the topic of e-voting, where there exists a wide literature. Chaum (1981) introduced the notion of mix-net as a tool for achieving anonymity in e-mail and in electronic elections. A mix-net consists in a sequence of servers, called mixes. Each server receives a batch of input messages and produces as output the batch in permuted (mixed) order. Such mix-nets are sometimes called mix cascades or shuffle networks. When used for voting, the input messages are the ballots of the voters. An observer should not be able to tell how the inputs correspond to the outputs. This property provides voter privacy in an electronic election. In Chaum’s original proposal, before a message is sent through the mix-net, it is encrypted with the public keys of the mixes it will traverse in reverse order. Each mix then decrypts a message before sending it on to the next mix. A modified version of the protocol was published later by Bicking et al. (1988). A new kind of receipt improves security by letting voters verify correctness of the election outcome, even though all election computers and records were to be compromised. The system preserves ballot secrecy, while improving access for voters, robustness, and adjudication, all at lower cost. Sako and Kilian (1995) propose another approach to e-voting based on re-encryption mix-nets (Park et al., 1994) and on proofs, used by voters to prove to the authorities the correctness of the votes they sent. Proofs may be interactive (e.g., classical zero-knowledge proofs) or non-interactive and simply attached to the vote. All mixes in this system have a unique private key for the El-Gamal encryption scheme. There exists a public key for an anonymous channel. Mixes produce encrypted ballots with proofs for users. During the voting stage, the voters choose their votes using an untappable channel and send them via decryption networks. This vote will be counted only after the mix posts a proof of correct decryption. This scheme is based on an ad-hoc recruitment, like for example the use of untappable channels for the transmission of data, which makes it little practical (Zwierko and Kotulski, 2007). Zwierko and Kotulski (2007) propose an agent-based scheme for secure electronic voting. The scheme is universal and can be implemented in a network of stationary and mobile electronic devices. The proposed system is based on an idea of an authentication protocol with revocable anonymity, which utilises a combination of Merkle’s puzzles (Merkle, 1978) and a secure secret sharing scheme (Pieprzyk et al., 2003). The Merkle’s puzzles provide anonymity and a secure secret sharing scheme is a method of group authentication. Both methods can also be used for the e-voting scheme to protect voter’s privacy and to create effective method of authorisation. The protocol, presented by Fujioka et al. (1993), is designed for large scale elections. The counter and voters communicate through an anonymous channel, the administrator uses a blind signature scheme so that each voter has a different digital signature and uses the commitment scheme to compute the ballot. The election protocols based on homomorphic encryption are described in various
papers (Acquisti, 2004; Cramer et al., 1997; Damgard et al., 2003). In the system proposed by Cramer et al. (1997) the authorities create a pair of shared private and public keys. Using these keys and the El-Gamal scheme, the voters can create their ballots: they encrypt their votes and produce a non-interactive proof of validity, with the zero-knowledge property. After checking the proofs from the voters, the coalition of honest authorities can combine all correct votes and utilise proofs to decrypt the product. In the result they obtain the final tally. The protocol proposed by Damgård and Jurik (2001) utilises the generalised Pallier’s cryptosystem. A more effective method of decryption and computing the result is presented by Cramer et al. (1997). Another system exploiting the homomorphic encryption scheme was proposed by Ogata et al. (1997) and improved by Abe (1998), Golle et al. (2002) and Park et al. (1994). During the initial stage the authority publishes the shared public key. Then, the voters register, compute and post their votes on a public, broadcast communication channel with memory, called bulletin board (Cramer et al., 1997). All votes are then sent through a re-encryption mixnet. After being checked, the count is made.

Some other approaches to electronic voting, also based on homomorphic encryptions, have been proposed by Acquisti (2004) and Juels et al. (2005). The system preserves the receipt-freeness property (and incoercibility, providing that the adversary does not have access to the registration phase), since a voter can generate a false receipt. Unfortunately, receipt-freeness and incoercibility have in the paper a high price in terms of verifiability and scalability. Also the usage of anonymous broadcast channel makes the scheme impractical, since it is hard to implement. All the above e-voting systems require complex mechanisms and ad-hoc infrastructures (e.g., mix-nets), making them not suitable for our purpose.

Chaum pioneered privacy-enhancing cryptographic protocols that minimise the amount of personal data disclosed. His work put forth the principles of anonymous credentials (Chaum, 1983, 1985; Chaum and Evertse, 1987), group signatures (Chaum and Van Heyst, 1991), and electronic cash (Chaum, 1983). These concepts have in common that some party issue a digital signature where the message signed includes information about the user (i.e., attributes). Subsequently, more efficient implementation of these concepts were proposed and a number of related concepts were introduced, including, group signatures (Ateniese et al., 2000; Boneh et al., 2004; Kiayias and Yung, 2006), e-cash (Brands, 1993; Camenisch et al., 2005; Frankel et al., 1998), anonymous credentials (Brands, 1995, 1997, 2000; Camenisch and Lysyanskaya, 2001, 2004), traceable signatures (Kiayias et al., 2004), anonymous auctions (Naor et al., 1999), and electronic voting based on blind-signatures (Fujioka et al., 1993). Many of these schemes use as building blocks signed attributes and protocols that selectively reveal these attributes or prove properties about them. The given implementations typically encode attributes as a discrete logarithm or, more generally, as an element (exponent) of a representation of a group element, resulting in protocols where the number of group elements transmitted and the commutations performed are linear in the number of encoded attributes.

An interesting approach to maximise privacy protection is to selectively disclose attributes within a credential, so that only the needed subset of properties is made available to the recipient of the credential. The best system currently available to allow partial disclosure of credentials relies on the use of the bit commitment technique, which enables users to commit a value without revealing it. Bit commitment has been used for zero-knowledge protocols (Goldreich et al., 1991; Brassard et al., 1988), identification schemes (Fiat and Shamir, 1987), and multiparty protocols (Goldwasser et al., 1987; Chaum et al., 1988), and it can implement Blum’s coin flipping over the phone (Blum, 1983). Although the idea of
selectively disclosing credential attributes is not new (Brands, 2000; Persiano and Visconti, 2000), this technique has never been thoroughly explored, especially in trust negotiations. The only work on this topic is from Jarvis (2003). This work focuses on selective disclosure of credentials during negotiations and provides a prototype implementation. The focus in the paper of Bertino et al. (2005), differently from Jarvis (2003), is to deeply analyse the impact of protected attribute credentials on trust negotiations, and devise new strategies to allow interoperability between users adopting various credential formats. Further, instead of using the bit-commitment technique the author adopt a multi-bit hash commitment technique for attribute encoding, as the length of attributes will likely be longer than one bit. The general protocol followed to issue credentials with protected attributes is briefly summarised in what follows. A credential requester first generates the set of attribute values for the credential. In order to create a credential with protected attributes the requester has first to create the corresponding private values to be used in place of the sensitive ones. These operations are performed for all the attributes of the credential that need to be selectively protected. Given a sensitive attribute a with value \(va\) the operations needed for its protection are:

- generate a random string \(r\)
- compute \(p = va \| r\), that the concatenation of \(va\) with \(r\)
- compute \(v = \text{hash}(p)\), generated by invoking a hash-function one way on \(p\).

Once the credential is ready, it is submitted to the credential authority, which verifies its content and the corresponding values, and finally signs it. During a negotiation, the credential can be sent by keeping secret the content of protected attributes. The disclosure of each private attribute is executed by sending the counterpart both \(va\), the original value, and \(r\), the random value, so that the receiver can compute \(va\) using the same hash function and verify the attribute validity.

Naor (1991) shows how a pseudorandom generator can provide a bit-commitment protocol and also analyses the number of bits communicated when parties commit to many bits simultaneously. Let \(m(n)\) be some function such that \(m(n) > n\). \(G : \{0, 1\}^n \rightarrow \{0, 1\}^{m(n)}\) is a pseudorandom generator. \(G_l(s)\) is used to denote the first \(l\) bits of the pseudorandom sequence on seed \(s \in \{0, 1\}^n\). \(B_i(s)\) is used to denote the \(i\)th bit of the pseudorandom sequence on seed \(s\). The user selects seed \(s \in \{0, 1\}^n\) and sends \(G_m(s)\) and \(B_{m+l}(s) \oplus b\). In the reveal stage, the user sends \(s\) and the verifier checks that \(G_m(s)\) is equal to the previously received value and computes \(b = B_{m+l}(s) \oplus (B_{m+l}(s) \oplus b)\).

The system of Holt and Seamons (2002) uses bit commitments to create selective disclosure credentials with a limited amount of data the holder must reveal. A selective disclosure credential has several attributes. When the user shows the credential to a verifier, she can choose to reveal only some of them. Credential sets accomplish this with the help of bit commitment that allows the user to commit to a value without revealing it. The user’s commitment is the output of a one-way function \(\text{oneway}()\) operating on the concatenation of her secret value \(s\) and a random string \(r\). The user first sends it to the verifier. If she chooses not to reveal the value, the verifier can’t determine what the value was. To reveal her secret, she sends \(s\) and \(r\) to the verifier who computes the one-way function and checks that the result equals the value sent previously by the user.

We observe that the approaches based on selective disclosure and bit commitment do not solve the problem investigated in our paper. Indeed, the secret used by a citizen to enable the disclosure of the chosen attribute would allow third parties to trace the citizen preferences in the different voting session, thus breaking unlinkability.
As a final remark, we observe that despite the wide related literature, our approach presents some interesting original features. Indeed, it implements in a new (lightweight) way the minimal kernel of security requirements (including secretness) taken from the field of e-voting, also enabling both voting analysis and continuous participation of citizens to the decisional government process. Observe that the last features are not supported by traditional e-voting systems and, thus, they place our paper in the domain of e-participation, where, in turn, no attention towards privacy is typically paid. In this sense, our paper proposes a new view merging the two fields of e-voting and e-participation.

3 An overview of the proposal

In this section, we briefly describe the scenario we have designed in our proposal. The e-voting protocol will be described in the next section. We assume that citizens may use a smart card embedding a certificate granted by any Certification Authority including only a unique numeric ID and a list of pairs \( \langle x, y \rangle \) where \( x \) is the attribute name and \( y \) is the obscured attribute value. This certificate includes information about the citizen in an obscured form, in such a way that the user may decide which information can be disclosed. Attributes encode standard information about users like personal data, but also more general information like job, qualification, marital status, etc. As usual, the certificate is a semi-structured document where the attributes are optionally included. For each attribute, its value is obscured by applying a one-way function using a key. A different key for each attribute is used. The keys are shared between the user and the Certification Authority. Thus, for a given attribute value \( A \) and a given key \( k \), we obscure the attribute value by computing \( g(A, k) \), where \( g \) is a one-way function. This means that it is unfeasible to compute \( A \) from the knowledge of just \( g(A, k) \). For function \( g \), we adopt the modular power function. The infeasibility of the computation of the discrete logarithm ensures us that the function is one way.

The solution is based on the usage of existing social networks. Citizens vote by using their social network profile. The e-voting infrastructure is implemented by exploiting, for each existing social network (actually, for the most important ones) a (large) number of profiles whose URL is of the form: http://www.social-network.org/poll_Y, where \( Y \) is a cardinal number. These profiles are managed by possibly different government entities. Each entity replicates its profile over the most common social networks. The only requirement we have for these super profiles is the service continuity. These profiles are called in our model credential providers, and play the role of granting credentials to voters they can spend in order to submit their vote to a TTP, responsible of generating the ballots for each e-voting. It is worth remarking that each credential provider belongs typically to a different party, in the chosen set. Indeed, we can imagine that the domain of credential providers is built by collecting a large variety of subjects, like public sector offices, postal offices, universities, schools, military subjects and so on, in such a way that we can easily reach a relevant number.

Recall that, our goal is to associate votes with some attribute values chosen by the voter. Indeed, the credential providers would able to incrementally relate information about the voter, as they know her identity. This way, the protocol would violate the confidentiality of the attribute certificate. By contrast, we want to relate votes to voters’ attributes without discovering to anyone whom these attribute refer to.

To do this, we include in the credentials a further obscuration of the attribute values of the certificate by means of a different key per attribute. Each credential provider shares these
A privacy-preserving e-participation framework allowing citizen keys with the user, but, obviously, it does not know the attribute values because operates only on already obscured values taken from the attribute certificate.

The credentials obtained by the voter contain a double obscuration of all the attributes of the voter, each with two keys, namely \( k \) and \( r \) (different for each attribute), in such a way that the knowledge of the product \( k \cdot r \) allows us to obtain the final obscuration starting from the plain value of the attributes. Indeed, for the chosen function \( g \), it holds that \( g(g(A, k), r) = g(A, k \cdot r) \).

This way, whenever the voter submits her vote to TTP, she decides which attributes to disclose, simply by including into the vote record the attribute value, say \( A \), and the product of keys \( k \cdot r \). Then, TTP for each chosen attribute \( A \), just has to compute the value \( g(A, k \cdot r) \) and to verify whether this value is included in the related credential.

The scenario is summarised in Figure 1. To avoid that the protocol is breakable by just one misbehaving credential provider, we use the common approach of replicating the responsibilities over a number of different independent parties (Cramer et al., 1997; Zwierko and Kotulski, 2007; Fouque et al., 2001; Hirt and Sako, 2000). In fact, the voter selects a suitable number \( t \) of credential providers on the basis of the value of her ID and asks them for the credentials necessary for the e-voting session. In Figure 1, we describe the different steps related to an e-voting session. First, the user receives the obscured certificate from a certification authority. Then, on the basis of her ID, the voter (we assume she has joined Facebook) computes four values (i.e., \( Y_1 = 456 \), \( Y_2 = 234 \), \( Y_3 = 768 \), and \( Y_4 = 986 \)) identifying the respective credential users (in this example, \( t = 4 \)). The TTP collects votes, verifies that they are admissible, and generates the ballots for each e-voting. The protocol ensures that the credential providers, even though may identify voters cannot link them to their vote, while TTP cannot identify voters but can only be aware about the attributes voluntarily disclosed by the voter.

Figure 1 The e-voting scenario (see online version for colours)

A basic feature of the system is that both credential providers and TTP store data regarding e-voting sessions only locally, so that the security of the protocol does not rely on the
trustworthiness of the social network providers. In other words, as far as e-voting information is concerned, the system is truly distributed. The only assumption we require (common in this context) is that no more than $t$ credential providers collude, where $t$ is a parameter of the system. Obviously, the number $t$ of contacted credential providers per voter is directly related to $t$. The detail of the protocol is shown in the next section.

4 The e-voting protocol

In this section, we describe how the e-voting protocol works. Throughout the paper we assume that the reader is familiar with concepts of digital signature, partially blind signature and modular arithmetic. Consider an e-voting session identified by $ID_v$. For the sake of presentation, we assume that a preference is expressed by reporting the number $i$ identifying the choice of the voter. For example, if a new law act is submitted for preliminary evaluation to citizens, rates from 0 to 10 could represent possible choices. Similarly, if the voting session regards the choice of a candidate among $k$ ones in a primary election, then the choice of the voter could be represented by a number from 1 to $k$. However, extending our technique to the cases in which preferences are given in a different way (for example, in the case of a primary election, by indicating the name of the candidate) is possible with no impact on the model.

The e-voting process involves the following basic entities:

- The Voter $V$. We describe how the protocol run for the voting done by one user. Clearly, the overall e-voting session involves many users, each running these steps
- A Certification authority CA granting attribute certificates to voters
- A set $\langle CP_1, \ldots, CP_c \rangle$ of $c$ special users, named credential providers, issuing the credential exploited by the voter to prove her authorisation to vote
- A trusted third party, say TTP, responsible of generating the certified ballots for each e-voting.

Our technique is parametric with respect to a value $t$. It is chosen in such a way that the likelihood that $t$ randomly selected users misbehave is negligible. This is a common assumption in this context (Cramer et al., 1997; Zwierko and Kotulski, 2007; Fouque et al., 2001; Hirt and Sako, 2000).

Now, we describe how the e-voting process proceeds. It consists of the following steps:

- Certificate issue. In this first step, CA generates the attribute certificate for the voter $V$ which contains $ID_V$ (i.e., a value that uniquely identifies each voter) and a list of $n$ associated attributes. All the attributes but $ID_V$ are obscured, in such a way that a third party cannot know the values of such attributes by accessing the certificate. In particular, for each attribute, its value is obscured by applying a one-way function using a key. A different key for each attribute is used. The keys are shared between the voter and CA. In detail, for a given attribute value $A$ and a given key $k$, we obscure the attribute value by computing $g(A, k)$, where $g$ is a one-way function. This means that it is unfeasible to compute $A$ from the knowledge of just $g(A, k)$. For function $g$, we adopt the modular power function $A^k \mod m$, where $m$ is a prime number greater than any possible $A$. In practice, $m$ can be set by assuming a realistic
Credential issue. If the above assumption is not applicable, we can use for each attribute a different module, which depends on the actual value of the attribute. In this case, the value of the module has to be saved in the certificate. Thus, CA selects a random vector of keys \((k_1, \ldots, k_n)\). Each attribute included in the certificate is a pair \((AN, g(AV, k_i))\), where \(AN\) is the attribute name and \(AV\) is the attribute value. Therefore, in the certificate, instead of the plain value \(AV\), only the obscured value \(g(AV, k_i) = AV^{k_i} \mod m\) is inserted. At the end of this operation, CA signs the certificate and sends it to \(V\) together with the vectors \((k_1, \ldots, k_n)\) and \((AV_1, \ldots, AV_n)\). We denote by \(C\) the so obtained certificate.

- **CPs identification.** In the first step, the voter, identified by \(ID_V\), has to select \(\ell = 2 \cdot t + 1\) among the \(c\) credential providers that will generate the credentials. The \(j\)th credential provider chosen by \(V\), say \(CP^V_p\), with \(1 \leq i \leq \ell\), is \(CP_j\), with \(j = \text{SHA-1}(|ID_V|) \mod c\). Specifically, the first credential provider is obtained by applying the hash function SHA-1 to the concatenation between the voter identifier \(ID_V\) and the number 1 (i.e., \(i = 1\)), and then by mapping the result to one of the \(c\) credential providers through the \(\mod\) operation. Note that the value \(j\) computed by SHA-1 corresponds to the number \(Y\) (discussed at the beginning of the previous section) completing the URL identifying the credential provider.

Furthermore, recall that all data regarding the e-voting procedure received by credential providers are stored only locally. This avoids that the social network provider could link all the information and violate the security measures of the system. In other words, we do not require any trust property to the social network providers. On the other hand, this is the reason why we require that credential providers rely on a local 24-hours-online server. Even this assumption, considering the type of involved parties, is realistic.

- **Credential issue.** In this step, the voter starts a connection with each \(CP^V_p\) (among the \(\ell\) ones). \(CP^V_p\) verifies that it has been correctly contacted by recomputing the function SHA-1 as done by \(V\) at the previous step. If this is the case, then \(CP^V_p\) generates the credential \(C^V_p\) allowing \(V\) to participate to the e-voting session. Otherwise, the connection is terminated. Before the generation of the credential, \(V\) sends the certificate \(C\) issued in Step 1 to \(CP^V_p\), together with a random vector \((r_1, \ldots, r_n)\), where, we recall, \(n\) is the number of the attributes in \(C\).

Then, \(CP^V_p\) generates a \(n\)-tuple of pairs \(AT = (AN_1, g(AV_1, k_1 \cdot r_1)), \ldots, (AN_n, g(AV_n, k_n \cdot r_n))\). Observe that the second element of the \(i\)th pair is the further obscuration of the \(i\)th attribute value by means of the random value \(r_i\), i.e., \(g(g(AV_i, k_i), r_i) = AV_i^{k_i \cdot r_i}\). We denote the attribute name \(AN_i\) by \(AT(i).name\) and the attribute value \(AV_i\) by \(AT(i).value\).

At this point \(CP^V_p\) is ready to construct the credential \(C^V_p\). It consists in the signature of the pair \((ID_{vs}, AT)\), where \(ID_{vs}\) is the identifier of the voting session.

- **Voting.** This step starts after the voter has collected one credential from each of the \(\ell\) credential providers. Such credentials are presented to TTP in order to obtain the possibility to vote. Observe that the communication between voter and TTP is not done through the social network, in order to avoid that the simple collusion between the social network provider and TTP might permit the identification of the voter by TTP.

Now, TTP performs the following tests on the received credentials:
• it checks authenticity and integrity of each credential and that the voting reference (i.e., $ID_{vs}$) in each credential coincide

• it verifies that in the past, no user has presented credentials issued from the same credential providers as the current voter for the same voting session (otherwise, it means that the voter is trying to repeat her participation to the same voting).

If both the tests succeed, then the voter is authorised to vote possibly disclosing some attributes.

Suppose now that $V$ decides to disclose $h$ attributes, with $h \leq n$. In this case, she must send to TTP the $h$-tuple of pairs $T = \langle (B_1, e_1), \ldots, (B_h, e_h) \rangle$, where $B_i$ is the value of a chosen attribute, say it the attribute $A_{x_i}$, and $e_i$ is the $i$th product $k_x \cdot r_x$, for $1 \leq i \leq h$. To verify that the voter choice is valid, it is necessary that TTP checks the consistence of $T$ with $AT$. In particular, given the function $f : \{1, \ldots, h\} \rightarrow \{0, \ldots, n\}$, defined as:

$$f(i) = \begin{cases} j & \text{if } \exists j \in [1, n] \mid AT(j).value = B_i^{e_i}, \\ 0 & \text{otherwise.} \end{cases}$$

TTP has to verify that $f$ is greater than 0 over all the domain $\{1, \ldots, h\}$. If this check fails, the vote is invalidated. Otherwise, TTP generates the ballot. The ballot consists in the partially blind signature of the quadruple $(ID_{vs}, \bar{r}, \bar{pr}, (AT(f(1)).name, B_{f(1)}), \ldots, (AT(f(h)).name, B_{f(h)})$, where $\bar{r}$ is a fresh 128-bit random sequence and $\bar{pr}$ represents the preference specified by the voter.

The values $ID_{vs}$ and $(AT(f(1)).name, B_{f(1)}), \ldots, (AT(f(h)).name, B_{f(h)})$ are unblindly signed, whereas $\bar{r}$ and $\bar{pr}$ are blindly signed. Finally, TTP stores the received credentials in order to detect a possible re-submission of the same credentials.

• **Ballot publication.** After the voter obtains the signed ballot, she unblinds it in order to obtain a new ballot still correctly signed by TTP but not linkable anymore to the voter. As usual, timing attacks are prevented by introducing an unpredictable delay before sending the new ballot back to TTP. The final ballot is thus $(ID_{vs}, r, pr, (AT(f(1)).name, B_{f(1)}), \ldots, (AT(f(h)).name, B_{f(h)})$.

Observe that, due to the presence of the tuples $(AT(f(1)).name, B_{f(1)}), \ldots, (AT(f(h)).name, B_{f(h)})$, the list of attribute names and values that $V$ has chosen to disclose is shown in the ballot.

At the end of the e-voting session, TTP verifies the signature of all received ballots and publishes valid ones. The presence of non identifying information about the voter enables the possibility to analyse citizens’ preferences in order to extract useful knowledge from them.

5 Implementation issues

In this section, we discuss the implementation of our proposal, analysing the data structures and algorithms to use. In our implementation, we use XML documents to handle the data exchange among the several actors (voter, certification authority, credential providers and
TTP). We define the structure of these XML documents by XML Schema (Fallside and Walmsley, 2004; Thompson, 2004).

We start with the definition of the structure of certificates, generated by the certification authority and issued to voters. It is modelled by the XML complex type certificateType (Listing 1). Each certificate has an identifier (serialNumber), the information about the certification authority (issuer), the validity period of the certificate (from start to end), an integer value identifying the voter (IDv), her/his public key (publicKey), the list of the properties associate with the voter (propertyList). Each property is a pair (name, value) (e.g., the name of a property could be ‘firstname’). Recall that, for privacy reasons, the value of all the properties in the certificate is obscured. Finally, the XML attribute key is not used in the properties of certificates.

Listing 1 The complex types certificateType, propertyListType and propertyType

```
<complexType name="certificateType">
  <sequence>
    <element name="serialNumber" type="integer"/>
    <element name="issuer" type="tns:issuerType"/>
    <element name="start" type="date"/>
    <element name="end" type="date"/>
    <element name="publicKey" type="hexBinary"/>
    <element name="IDv" type="integer"/>
    <element name="propertyList" type="tns:propertyListType"/>
  </sequence>
</complexType>

<complexType name="issuerType">
  <sequence>
    <element name="commonName" type="string"/>
    <element name="organizationalUnit" type="string"/>
    <element name="organization" type="string"/>
    <element name="country" type="string"/>
  </sequence>
</complexType>

<complexType name="propertyListType">
  <sequence>
    <element name="property" type="tns:propertyType" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="IDvs" type="integer"/>
</complexType>

<complexType name="propertyType">
  <attribute name="name" type="string" use="required"/>
  <attribute name="value" type="hexBinary" use="required"/>
  <attribute name="key" type="hexBinary"/>
</complexType>
```

Now, we introduce the XML complex type called credentialType, which represents the credential generated by the credential provider and issued to the voter. Its structure is reported in Listing 2. In order to avoid linkability, a credential does not include an identifier. It includes:

1 the integer value identifying the credential provider (IDp)
2 the integer value identifying the voting session (IDvs)
3 the list of the properties associate with the voter (AT).
Recall that these properties have the same name as the properties of the certificate, but their values have been furthermore obscured by a random key (see Step 3 of Section 4).

**Listing 2**  The complex type `credentialType` in the XML Schema `voting`

```xml
<complexType name="credentialType">
  <sequence>
    <element name="IDp" type="integer"/>
    <element name="IDv" type="integer"/>
    <element name="AT" type="tns:propertyListType"/>
  </sequence>
</complexType>
```

As for the ballot generated by TTP, its structure, called `ballotType`, is reported in Listing 3. The ballot stores a reference to the voting session (`IDv`), and the blindly signed value (thus, the XML attribute `partiallyBlindType` is equal to true) of a 128-bit random sequence (`r`) and of the preference specified by the voter (`pr`). Moreover, the voter includes in the ballot the properties to disclose (`disclosedPropertyList`): for each property, the XML attribute `value` is provided in plaintext (i.e., it is not obscured) and the XML attribute `key` contains the secret necessary to transform this value from obscured to plaintext, in order to verify the correctness of this disclosure.

**Listing 3**  The complex types `ballotType` and `partiallyBlindType` in the XML Schema `voting`

```xml
<complexType name="ballotType">
  <sequence>
    <element name="IDv" type="integer"/>
    <element name="r" type="tns:partiallyBlindType"/>
    <element name="pr" type="tns:partiallyBlindType"/>
    <element name="disclosedPropertyList" type="tns:propertyListType"/>
  </sequence>
</complexType>
<complexType name="partiallyBlindType">
  <simpleContent>
    <extension base="hexBinary">
      <attribute name="partiallyBlind" type="boolean"/>
    </extension>
  </simpleContent>
</complexType>
```

The data structures described above are contained in the XML root element `voting` (Listing 4).

After describing the data structures used in the implementation of our protocol, we discuss the algorithm performing the voting step (see Section 4), which is the only one requiring a more detailed explanation. Its pseudo-code is shown in Algorithm 1.

The algorithm receives as input the XML document, say `voting.xml`, presented to TTP by the voter. This document contains the certificate issued by the certification authority, the credentials collected by the voter, and the ballot that TTP will sign if the check of the presented credentials succeeds. The output is an XML document, say `ballot.xml`, which is the signed ballot. Observe that, in our pseudo-code we use XQuery expressions (W3Schools.com, 2013) to extract and manipulate data from the XML documents and SQL expressions for data insertion to and querying from a database called DB, which supports our implementation.
The algorithm proceeds as follows. Each credential presented by the voter is analysed. The
XQuery expression at line 2 returns all the XML elements `credential` children of the
root `voting` in the document `voting.xml`. First, the authenticity and integrity of the
signature of the credential is verified by using the method `checkSignature` (line 3). A
fake credential is skipped and no further analysis is carried out – this is implemented by the
`continue` statement (line 4). Then, it is checked that the credential has been issued from
the credential provider for the voting session declared by the voter in the ballot (line 6). The
method `number` receives as input the XML element or attribute and returns its value.

To detect a possible re-submission of the credential, it is checked by an SQL query
(line 9) whether the credential is included into the table `UsedCred` of the database DB
storing references to all credentials received in the past from TTP. This table consists of
two columns, the former identifies the credential provider issuing the credential, the latter
identifies the voting session. If the table already includes this credential, it is skipped.
Otherwise, the reference to the current credential is inserted into the table of the already
presented credentials. The variable `cont` is incremented (line 14) each time a credential is
valid (i.e., it succeeds all tests).

Now, it is verified (lines 15–29) the consistence of the attributes that the voter decides
to disclose (`/voting/ballot/disclosedPropertyList/property`) with the
attributes contained in the credential (`/voting/credential/AT/property`). An
inconsistence occurs if a disclosed property either

- it appears in the credential with a different value (`wrong = true`)
- it does not appear in the credential (`found = false`). The variables `wrong` and
  `found` are used both to detect such cases and, as optimisation, to break the `for` loop.

The method `decrypt` receives as input two XML attributes that represent the obscured
value of the property and a key and returns an XML element with the plaintext value of the
attribute if the key is correct, null otherwise. If an inconsistence is found, then the property
incorrectly disclosed from the voter is removed from the ballot (line 27).

Finally, if at least \( \bar{t}/2 + 1 \) credentials are valid, TTP proceeds by blindly signing the
ballot (line 32). We recall that \( \bar{t} = 2 \cdot t + 1 \), where \( t \) is a parameter of the system defined at
the end of Section 3.
The output of the algorithm is an XML document ballot.xml containing the ballot and its partially blind signature.

6 Security analysis

In this section, we prove that our proposal is resistant to the numerous attempts that an attacker can carry out to break our protocol. In our analysis, we assume that the cryptographic hash function and the partially blind signature adopted in the implementation are secure.

The first case we analyse is that of the attacker who tries to guess the real value of the attribute $A$ from its obscured value saved in the certificate as $A' = A^k \mod m$. In this case, the attacker must find the inverse discrete logarithm of $A'$, which is unfeasible. Even when the attacker can obtain more credentials containing the same attribute $A$, the attack fails. Indeed, the $i$th credential contains the obscured value $A^{(k \cdot r_i)} \mod m$, where $r_i$ is a salt, and an attack on this product is unfeasible thanks to the bit length of the salt. Moreover, the salt is used to break a possible link between the certificate and the credentials issued to the same voter. Indeed, the voter $ID$ is not included in the credential and any attribute $g(AV_i, k_i)$ in the certificate is transformed into $g(AV_i, k_i \cdot r_i)$. Thanks to the further obscuration performed by $r_i$, there is no way, without the knowledge of this random value, to link the credential to the attribute certificate and to the voter.

As for the link between voters and their preference in the ballot, it is prevented by the use of the partially blind signature (see line 34 of Algorithm 1) because this signature produced by TTP on the ballot hides the preference score.

Concerning the uniqueness of the vote, in case the attacker tries to replay the same credentials, the double vote is detected by TTP that stores all presented credentials (see line 12 of Algorithm 1). We consider now the possibility that two voters $V_1$ and $V_2$ share the same set of credential providers due to the collision of the hash function SHA-1. The result is that they are considered the same voter and the second vote is annulled because it is detected as duplicated vote. However, the probability of this event is negligible in a realistic scenario. For example, as the number of possible different sequences of credential providers is $c!/(c-t)!$ (we recall $c$ is the number of credential providers and $t = 2 \cdot t + 1$), for the realistic values $t = 21$, $c = 200$, and even hypothesising an unrealistically high number of $10^{12}$ voters, the probability of this event is less than $10^{-20}$.

Consider now the verifiability of votes. Each vote is identified by $r$ (known only by the voter) in the published ballot list and thus voters can verify their preferences. In this case, we analyse the possibility that two voters generate the same 128-bit sequence $r$. With consideration similar to that done for the birthday attack (Coppersmith, 1986; Girault et al., 1988; Kirchner, 2011), this probability is $p(u; D) \approx 1 - e^{-u^2/(2 \cdot D)}$, where $u$ is the number of users expressing their preference and $D$ is the domain of $r$. Assuming again a number of users $u$ equals to $10^{12}$ (in the worst case), such a probability is less than $10^{-15}$.

The next security feature we analyse is uncloneability, which requires to detect any attempt to create a fake ballot. This is guaranteed because any ballot has to be signed by TTP and thus it cannot be modified. Obviously, it cannot be duplicated thanks to the presence of the bit-sequence $r$ identifying the ballot, according to the previous probability considerations.
Concerning the possibility that two obscured values, say $g(AV_1, k_1 \cdot r_1)$ and $g(AV_2, k_2 \cdot r_2)$, collide in $AT$, the probability of this event is negligible thanks to the randomness of $r_1$ and $r_2$ assuming that the number of bits of such random values is sufficiently large. According to this observation, even though from a formal point of view the definition of the function $f$ does not allow us to guarantee that $f$ is deterministic, from a practical point of view $f$ returns always a unique value when it is defined.

### Algorithm 1 Voting

**Input**  
$voting.xml$: an XML document  
**Output**  
$ballot.xml$: an XML document  
**Variable**  
$cont$: an integer  
$wrong$: a boolean  
$found$: a boolean  
  1: $cont := 0$  
  2: for $c$ in doc("voting.xml")/voting/credential do  
  3: if checkSignature($c) = false then  
      continue  
  4: end if  
  5: if number($c/ID_c) = number(doc("voting.xml")/voting/ballot/  
      disclosedPropertyList/@ID_c) then  
      continue  
  6: end if  
  7: if (SELECT * FROM BD.BurnedCred  
      WHERE $ID_p = number($c/ID_p) AND $ID_c = number($c/ID_c)) != null then  
      continue  
  8: else  
      INSERT INTO DB.BurnedCred VALUES (number($c/ID_p), number($c/ID_c))  
  9: end if  
  10: end for  
  11: $wrong := false$  
  12: for $dp$ in doc("voting.xml")/voting/ballot/disclosedPropertyList/property do  
  13: $found := false$  
  14: for $p$ in doc("voting.xml")/voting/credential/AT/property do  
  15: if $dp/@name = $p/@name then  
      $found := true$  
  16: if $found AND $dp/@value != decrypt($p/@value, $dp/@key)$ then  
      $wrong := true$  
  17: break  
  18: end if  
  19: end if  
  20: end for  
  21: if $wrong OR !found then  
      REMOVE $dp$  
  22: end if  
  23: end for  
  24: end if  
  25: end for  
  26: $wrong := false$  
  27: for $dp$ in doc("voting.xml")/voting/ballot/disclosedPropertyList/property do  
  28: $found := false$  
  29: end for  
  30: end for  
  31: $wrong := false$  
  32: if $cont > t/2$ then  
      $<voting>$  
  33: blindSignature(doc("voting.xml")/voting/ballot)  
  34: $<voting>$  
  35: end if  
  36: end if

Consider now the property of robustness and recall that the basic assumption in our proposal is that at most $t$ users misbehave during the whole evaluation process. Because the application of the hash function SHA-1 at Step 2 returns a pseudo-random value depending on the (identifier) voter, this allows us to state that the credential providers selected by the voter can be considered randomly chosen. Thus, the unfair behaviour of at most $t$ credential providers is detected, because, among the $\tilde{t} = 2 \cdot t + 1$ credentials provided by the voter, at least $t + 1$ of them must be correct. As a consequence, fake credentials are detected because they are in the minority.
Besides security aspects, we finally observe that our solution presents a good scalability, because the number of users involved in the generation of a single ballot is independent of the overall number of users. In particular, the preference score involves $2 \cdot t + 1$ credential providers.

7 Conclusion

The participation of citizens to the governance of the community they belong to is one of the key factors of the development of e-democracy. An important role that citizens should have in this context is the possibility to express their opinion about public decisions, including here any case of expression of preferences not reaching the formality of official elections. But the main basic properties of an e-voting system are desired also in these more informal cases, in order to guarantee their significance. In this paper, we have proposed a lightweight e-participation framework and we have addressed the main implementation issues. Our framework relies on the use of existing social networks and allows the voter to graduate the privacy level of the vote. In particular, citizens may decide, whenever submit a preference, to reveal some non-identifying personal certified attributes to link to the vote. This way, a-posteriori analysis on ballots can be done without threatening citizens’ privacy. The e-participation framework is oriented to all those processes aimed at collecting opinions, preferences, evaluations of citizens, which assume a very important role in the e-democracy model, since represent the concrete way to adapt government decisions to the real expectations of citizens.

The result we have obtained is a fair compromise between the secreteness of the vote and the necessity of government parties to conduct analyses on the collected opinions, in order to relate them to various types of information describing the inquired population. The solution shows also good features of feasibility since it does not require complex ad-hoc infrastructures by exploiting pervasive and user-accepted media (i.e., social networks). The security analysis also shows that all the basic properties of an e-voting system are satisfied and that a correct utilisation of our extended notion of secreteness does not invalidate the anonymity of the voters.

Acknowledgements

This work has been partially supported by the TENACE PRIN Project (n. 20103P34XC) funded by the Italian Ministry of Education, University and Research and by the Program “Programma Operativo Nazionale Ricerca e Competitività” 2007–2013, Distretto Tecnologico CyberSecurity funded by the Italian Ministry of Education, University and Research.

References

A privacy-preserving e-participation framework allowing citizen 203


A privacy-preserving e-participation framework allowing citizen


