Towards Fair Indictment for Data Collection with Self-Enforcing Privacy

Mark Stegelmann

Centre for Quantifiable Quality of Service in Communication Systems*,
Norwegian University of Science and Technology, Trondheim, Norway
mark.stegelmann@q2s.ntnu.no

Abstract. Recently, multiple cryptographic schemes for data collection with self-enforcing privacy were proposed by Golle et al. The schemes allow participants of electronic polls to prove a pollster's guilt if he distributes responses. Introducing punitive damages for such misbehaviour creates incentives for a pollster to protect the respondents' privacy. To achieve fairness, a proof must be feasible if and only if a pollster indeed leaked information. This paper analyses the scheme proposed for self-enforcing privacy with no release of data. Neither parameter publication nor cooperative indictment have been defined up to now. We show that both are of key importance to ensure fairness and describe potential attacks of a malicious pollster. After a detailed analysis, we propose two extensions preventing such actions. In addition, a possibility for the pollster to gain an unfair advantage in the basic scheme is identified and according checks put forward.

1 Introduction

When trying to conduct electronic polls about sensitive topics, pollsters face a specific challenge. Participants will be reluctant to provide accurate responses if they fear that the pollster might distribute their answers to other parties such as insurance companies or marketeers. Although some inaccurate answers are to be expected in polling scenarios, only a certain amount will be tolerable when looking at accumulated results. Since a pollster is interested in meaningful results, non-disclosure might be considered an implicit goal of him as well. However, it is usually not possible for respondents to conclusively assess a pollster's trustworthiness. In other words, no trust between participants and pollster can be inferred. To the same extent, it is assumed to be infeasible to find a trusted third party that can be relied on by all respondents to sufficiently protect their privacy. In order to address this challenge, several cryptographic schemes for data collection with self-enforcing privacy have been proposed [1].

The remainder of this document is structured as follows. We continue this section by giving a short overview of data collection with self-enforcing privacy

* "Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence" appointed by The Research Council of Norway, funded by the Research Council, NTNU and UNINETT. http://www.q2s.ntnu.no

and the scheme with no release of data as proposed by Golle et al. [2]. Section 2 provides our detailed analysis of the scheme's fairness. We identify a way for pollsters to gain an unfair advantage in the scheme and put according checks forward. In addition, we examine the effects of parameter publication on the scheme's fairness in Sect. 3. This leads to our proposal of two solutions in Sect. 4 and Sect. 5 respectively. In Sect. 6 we elaborate on the issue of cooperative indictment before concluding with Sect. 7 by pointing out future research directions.

1.1 Data Collection with Self-Enforcing Privacy

Data collection with self-enforcing privacy for electronic polls proposed by the authors of [1] is based on two key concepts. First, making privacy violations publicly provable and second, introducing consequences for such misbehaviour. In other words, the goal is to convincingly align the interests of a pollster with the respondents' desire to keep their sensitive answers from being distributed.

Incentives for a pollster to not disclose information may for instance be that he would have to forfeit a bounty or face legal consequences when proven to have revealed responses. However, a proof of a privacy violation must be feasible if and only if a pollster indeed was the source of the leaked information. That is to say, a sufficient level of *fairness* between the involved parties has to be achieved. The aspect of fairness can be seen as one of the key differences to for instance digital watermarking techniques such as those given in [3]. Although watermarking introduces additional information as well, it does not allow proofs on which party leaked information.

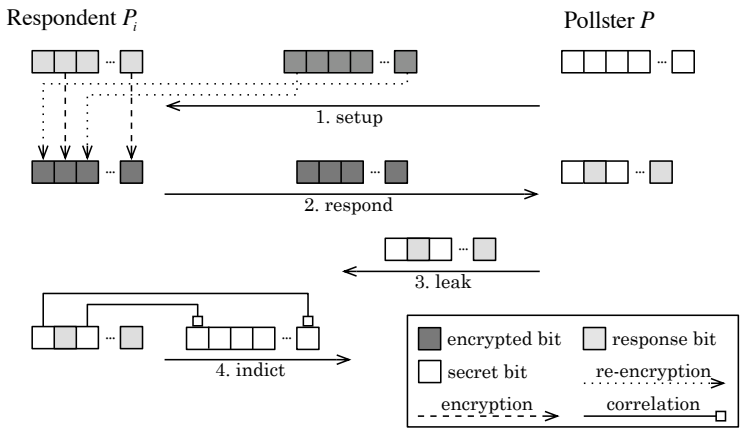

Fig. 1. Basic SEP overview

Figure 1 shows a conceptual overview of the first scheme with no release of data [2] and its four key phases. We will refer to it as *Basic SEP*. Both the

respondents and the pollster are depicted in the figure. The former are denoted by P_i, the latter by P. Message flows between them are visualised by horizontally oriented arrows. Note that for comprehensibility reasons some protocol details have been omitted in this illustration. They will be described in detail during the analysis in Sect. 2.

The scheme requires a homomorphic public-key cryptosystem that is semantically secure under re-encryption such as the ElGamal cryptosystem [4]. During Basic SEP's setup phase the pollster generates a random secret, encrypts all bits of this secret individually with his public-key, and publishes the respective encryptions to the respondents. In addition, he puts a *bounty* on the successful recovery of the secret. Due to not knowing the pollster's private key and the cryptographic security of the public-key cryptosystem, the respondents are unable to decrypt the secret. They can however in turn bitwise intertwine their answers, also encrypted with the pollster's public-key, with re-encrypted parts of the secret. Since the cryptosystem is assumed to be semantically secure under re-encryption the pollster is unable to distinguish regular encrypted submission from re-encrypted special bits. As a result, the latter, so-called *baits*, will inevitably be decrypted by him when trying to recover respondents' answers. If eventually sensitive, decrypted data consisting of real responses and secret bits are leaked by the pollster, respondents can indict him to claim the bounty.

The prospect of a bounty may in addition create incentives for an opportunistic third party called *bounty hunters* to actively try to uncover privacy violations. Respondents may collaborate with each other or with one or several bounty hunters during indictment. Note that Basic SEP does not address the issue of how leaked information get known to respondents. Instead, it focuses on providing the necessary groundwork to allow fair proofs if this happens.

2 Security Analysis

We now describe our security analysis of Basic SEP. As explained in Sect. 1 a self-enforcing privacy data collection scheme needs to fulfil two key requirements. It needs to make privacy violations provable and to introduce consequences for such misbehaviour. All the while, fairness as defined in Sect. 1 has to be ensured. Although additional properties such as high poll accuracy or safe result publications can be desirable in some scenarios, we limit our analysis on the two basic properties.

As mentioned before, Golle et al. assume a scenario with multiple respondents and a single pollster for their scheme [2]. No trust relationships between respondents and pollster and no trusted third party exists. The pollster is assumed to have published the public parameters for a homomorphic public-key cryptosystem that is semantically secure under re-encryption. For the ElGamal cryptosystem these public parameters are a group G and a generator $g \in G$ of a multiplicative subgroup G_q of order q in which the Decisional Diffie-Hellmann problem is hard. With x being the private key the public-key is $y = g^x$. We will denote encryptions (g^r, my^r) of a message m shorthand by $E(m)$, decryp-

tions of such a ciphertext by $D((K, M))$ can then be calculated using M/K^x. Note that re-encrypting a ciphertext $(K, M) = (g^r, my^r)$ is possible with just the knowledge of the cryptosystem's public parameters. The according re-encryption function $R((g^r, my^r))$ returns a ciphertext (g^{r+s}, my^{r+s}) that decrypts to the initially encrypted m for some random s element of the subgroup of order q. Basic SEP also makes use of discrete logarithm proof systems and proofs of correct decryption [5, 6]. We adopt the notation of $PCD(E(m) \rightsquigarrow m)$ to refer to a protocol instance proving correct decryption of a ciphertext $E(m)$ to m. The data submitted by respondents is assumed to be protected by a layer of symmetric-key encryption to protect against the used cryptosystem's weakness regarding chosen-ciphertext attacks.

2.1 Property I: Provableness of Privacy Violations

In order to make privacy violations provable Golle et al. propose that the pollster publicly commits to some secret. He has to publish this commitment in such a way that parts of it can be—undetectably to him—introduced into the respondents' regular submissions and thus will unknowingly be decrypted by him when trying to recover answers. All of the above has to be done in such a way so that the scheme's fairness is not compromised. This means on the one hand that respondents must be able to verify the correct execution of all steps. On the other hand it means that they must not be able to wrongly indict the pollster. The setup, the submission, as well as the indictment phase are thus relevant for this property.

Table 1 depicts the protocol flow of the setup phase which we derive from the textual description the authors give in [2]. It starts with the pollster randomly choosing a k-bit secret β with k being a security parameter determined by him. He publishes encryptions of the individual bits taken to the power of g and

Respondent P_i		Pollster P
g, E		g, E, D
	$\xleftarrow{E(g^{b_1}),\ldots,E(g^{b_k})}$	$\beta \leftarrow b_1 \ldots b_k \in_R \{0,1\}^k$
	$\xleftarrow{\forall_{i=1}^{k}(PCD(E(g^{b_i}) \rightsquigarrow g^0) \vee PCD(E(g^{b_i}) \rightsquigarrow g^1))}$	
	$\xleftarrow{g^\beta}$	$E(g^\beta) \leftarrow \prod_{i=1}^{k} (E(g^{b_i}))^{2^i}$
	$\xleftarrow{PCD(E(g^\beta) \rightsquigarrow g^\beta)}$	
	$\xleftarrow{\text{put bounty on recovery of } \beta}$	

Table 1. Basic SEP setup phase

proves with a disjunctive discrete logarithm proof system that each encryption decrypts to either g^0 or g^1. After that he calculates $E(g^\beta) = \prod_{i=1}^{k} (E(g^{b_i}))^{2^i}$, publishes g^β, and proves with a proof of correct decryption that the encrypted

value indeed decrypts to g^β. In the last step of the setup phase, he puts a bounty on the recovery of β.

Analysing the described steps reveals that they enable respondents to verify several things. First, they know that if they submit encrypted answers of the form g^b with b being a response bit then their submissions will decrypt to the same form as secret bits. Second, the respondents learn g^β. Due to the assumed multiplicative homomorphism, respondents are able to calculate the encryption of g^β without having to rely on the respective decryption. They can use the same equation as the pollster for this. The according proof of correct decryption assures them that the g^β provided to them by the pollster is indeed the correct decryption of the sequential composition. This means that if they learn the individual bits of the secret, they can claim the bounty.

However, if the pollster does not leak bits, neither information about the private key of the pollster nor the decryption of β is revealed. If we assume the respondents to be polynomial time bounded the computational complexity of recovering β without leaked bits is computationally hard since it would require solving the discrete logarithm problem. Thus, they are unable to decrypt secret bits and cannot leak information themselves to wrongly accuse a pollster afterwards. Observe that while the hiding property of the commitment is as just mentioned computational, the binding property is absolute.

The goal of the submission phase in turn is for the respondents to be able to send indistinguishable encryptions of bits and baits to the pollster. This means that respondents can choose between two alternatives when asked to submit a response bit. As shown in Tab. 2 they can either opt to submit a real bit b by sending an encryption of g^b or to send a bait. To do the latter, they let r be

Respondent P_i		Pollster P
g, E, R, b		g, E, D
submitting either bit b:		
$s \leftarrow E(g^b)$	$\xrightarrow{s}$	$D(s) \overset{?}{\in} \{g^0, g^1\}$
		$b' \leftarrow log_g(D(s)) = b$
or bait:		
$r \in_R \{1, \ldots, k\}$		
$s \leftarrow R(\underbrace{t}_{E(g^{b_r})})$	$\xrightarrow{s}$	$D(s) \overset{?}{\in} \{g^0, g^1\}$
		$b' \leftarrow log_g(D(s)) = b_r$

Table 2. Basic SEP submission (single bit)

a random index for the secret β and submit a re-encryption of the bit at the respective position. Respondents cannot gain an unfair advantage by deviating from the described behaviour in this scheme. The reason for that is that the only way for them to get to know the decryptions is when a pollster leaks the data since no results are published. The pollster in turn will only accept responses that decrypt to the form g^b with $b \in \{0, 1\}$.

The used cryptosystem is semantically secure under re-encryption and submission bits are encoded in the same form as secret bits. Consequently, the pollster is unable to distinguish encrypted bits from re-encrypted baits by comparing them with encrypted baits or by looking at their form after decryption. However, the proposal by Golle et al. does not guarantee that bits are indistinguishable from baits to the pollster, as we will show now.

A malicious pollster who wants to be able to distinguish bits from baits can choose β to only consists of 0s or 1s respectively. He does so instead of using random bits when setting up the poll parameters as shown in Tab. 1. Let us assume that he selects 0. When submitting a bait according to the protocol given in Tab. 2 respondents will as a result always re-encrypt values that decrypt to 0. A truthful answer bit in contrast may be either 0 or 1. Bits of value 0 will also result in encrypted values of 0. If respondents however answer with a 1 the pollster can identify a truthful response and be sure that it is not a bait. With prior knowledge of the to be expected distributions of answers and the ability to formulate the poll questions accordingly, the pollster can optimise the amount of learned bits. Note that this possibility to learn bits does not violate the property of semantic security under re-encryption of the chosen cryptosystem. The pollster is only able to distinguish bits from baits only after they have been decrypted.

This behaviour can be detected by the respondents. It can even be observed before submitting any bit. What is required is the addition of tests for the two respective forms of β being either 0 or 2^k after having received the poll parameters. However, a malicious pollster may in turn try to evade these checks by deviating from the two basic forms. He still can achieve a probabilistic chance of learning truthful answers. More certainty in this case however also means easier detection and thus an increased chance of having to forfeit the bounty. Thus participants need to check for those cases as well. Although the malicious behaviour can be detected, it needs to be actively checked for in addition to the original scheme's protocol. Otherwise, the pollster may gain an unfair advantage.

2.2 Property II: Consequences for Misbehaviour

In order for Basic SEP to be self-enforcing it needs to provide reasons for abstaining from misbehaviour. The bounty and the chance of having to forfeit it when a privacy violation is proven create these incentives. This means that a pollster needs to put a bounty on the successful recovery of the secret and it needs to be possible to claim the bounty when a privacy violation is proven. Accordingly, the setup and the indictment phase are important for the analysis of this property.

As shown in Tab. 2, the setup phase concludes with the pollster putting a bounty on the recovery of the bait. Golle et al. however neither specify how this step nor how the indictment phase is to be designed. It is clear that in this scheme the verification of an indictment's validity does not need to rely on the cooperation of the pollster. If β is known, the verification can be done efficiently without the secret key. If this would not be the case, a pollster could refuse to partake in his own indictment as it is the case for another scheme [7].

The unspecified publication process of the scheme's parameters during the setup phase is of central importance for the fairness of the scheme. To see that, assume a pollster communicating different parameters to respondents. New respondents could always opt not to take part in polls if parameters get modified to intolerable values. Yet, a pollster could, e.g., choose a different secret for every participant. As a result, respondents would have difficulties indicting him. If they did not submit enough baits to do so alone it would be computationally infeasible to them as explained in Sect. 2.1.

But even if the pollster did not change parameters, depending on the number of submitted bits and baits single respondents may not gain enough information to indict the pollster on their own. Because baits lower a poll's accuracy neither should the scheme depend on nor should it give incentives to submitting large fractions of them. Thus, respondents should be able to collaborate with each other or with some bounty hunter in order to indict a malicious pollster. The challenge of implementing this cooperative indictment without endangering the scheme's fairness is discussed in Sect. 6.

3 Fair Poll Parameter Publication

We focus on the issue of poll parameter publication first. The cryptosystem's parameters, the commitment to the secret and according commitment parts, and the statement to forfeit a bounty need to be communicated to respondents in a way that does not imperil fairness.

The latter statements to forfeit a bounty can be constructed in such a way that repudiation attempts are not possible. Requiring the statements to be digitally signed by the pollster while assuming a public-key infrastructure (PKI) to be in place would be one way. Yet, other parameter modifications can affect the scheme's fairness. If a pollster can change the cryptosystem's parameters or the secret, a justified indictment can be, as explained, rendered harder or even infeasible. By changing the secret the pollster can keep the amount of secret bits respondents can potentially learn from leaked responses below a defined threshold. Even if it is assumed that a pollster cannot alter published parameters at will, he may still publish different sets of parameters. The scheme proposed by Golle et al. provides no way to indict the pollster in case of either behaviour.

Let us analyse the extent of this issue. Assume that the respondents are using the method of Pollard [8] when trying to recover the secret, as suggested in [2]. They then can find β in time $2^{(k-l)/2}$ with $l < k$, k being the security parameter, and l the amount of bits known to them. The pollster can calculate the number n of participants that can be given the same secret without risking the bounty. Let r be the number of response bits being requested from each participant. Then the statement

$$2^{(k-nr)/2} > 2^x \tag{1}$$

expresses that the time required by respondents to recover the secret should exceed an amount of time 2^x which is assumed to be unreasonable for them.

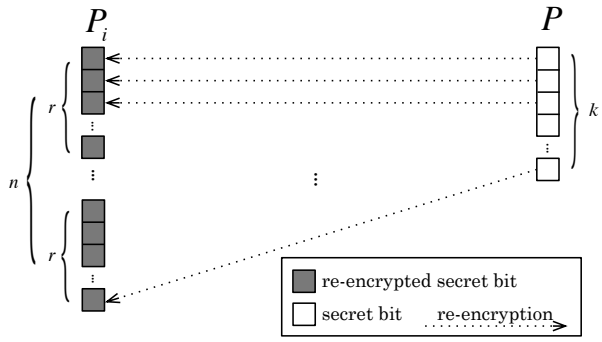

Fig. 2. coordinated collaborative indictment

Note that this is a worst case scenario calculation as depicted in Fig. 2. The pollster assumes that all participants of one group spend all their bits in a coordinated fashion meaning that $l = nr$. In other words, they try to uncover each single bit of the secret only once. Although regular respondents are unlikely to submit baits in such a coordinated fashion, when assuming opportunistic bounty hunters a malicious pollster has to bear in mind this scenario. Equation (1) can be transformed to

$$n < \frac{k - 2x}{r} \tag{2}$$

as well, which allows the pollster to determine an upper bound for the number of participants he may give the same secret to.

Consider a numerical example of k being 160 for a poll asking for $r = 13$ bits with x being 60. Then n needs to be less than 3. It is also possible to calculate the maximum number of responses a pollster can safely leak if he changes the secret for every respondent. If we let k and x be as defined before and n be equal to 1 then r has to be less than 40 bits. In general, a pollster can choose k and r in such a way that $k/r > 1 + 2x$ in order to be able to publish data without risking his bounty.

In addition, it is possible for him to hamper such a coordinated effort. First, he can introduce random noise into leaked responses to increase the number of required coordinated respondents. In presence of noise respondents need to rely on some technique such as a majority vote to find out the real values of secret bits. Second, he can try to detect such response sets by looking at the distribution of the bits. In contrast to regular responses, it is likely to be close to the one of the secret bits since participants cannot know which encrypted bits they re-encrypt.

The important question to discuss thus is if detecting poll parameter modifications is a sensible approach. It is possible to check for polls querying similar questions. Such checks can however be evaded by adding superfluous questions, syntactically modifying them etc. Besides the challenge of objectively estimating poll similarity, there may be valid reasons to ask identical questions in different

questionnaires. Basic information such as age or sex can be interesting for many polls. Therefore, in the following we examine possibilities of abstracting from polls and imposing limits on pollsters themselves.

4 Preventive Poll Parameter Limitation

We begin by describing our preventive approach. It relies on the existence of a third party T trusted to enforce a rate limiting mechanism and can be divided into the following two phases:

1. **parameter publication:** P requests poll parameter signature by T
   if rate limit for P not exceeded: T provides signed parameters to P
2. **submission:** P produces signed parameters to P_i
   if signature valid and parameters acceptable: P_i proceeds as in Tab. 2

The first step of poll parameter publication includes the protocol given in Tab. 1 with the exception of T taking the place of P_i. That means that P has to setup the poll with T which returns signed parameters if the protocol is successful and the rate limit was not exceeded.

The rate limiting mechanism can be implemented to be explicit or implicit. An explicit limit can be placed on the number of poll parameter creations and modifications allowed for a pollster in a given time frame. An implicit limit can be realised by requiring, e.g., a fee from pollsters for the signing of parameters. Although theoretically possible, we do not distinguish between newly created and modified poll parameters because of the, as discussed, similar effects on fairness. The goal of both the explicit and implicit limit is to preserve the scheme's fairness. Thus, their according parameters meaning number of changes and amount of money respectively have to be chosen accordingly. Equation (2) can help in determining these values for concrete instances.

The implicit approach has two important advantages over the explicit one. First, it does not require the storage of any prior records. This also means that there is no need for synchronisation of any kind when extending the concept to allow multiple signing trusted third parties. Second, and more importantly, it is not necessary for T to be able to verify the identity of the pollster. Such a verification is needed for the explicit mechanism in order to prevent denial of service attacks against the pollster by malicious impersonators. Both of the described approaches are preventive. In other words, neither one establishes a new indictment process.

When looking at the assumptions, a third party trusted in the above described way could be challenged to be a too strong and restrictive assumption, removing the benefits of the decentralised nature of the scheme. However, the overall goal of fairness must be guaranteed and a third party enforcing a rate limit is still a less strong requirement than the otherwise required party that needs to be trusted unconditionally to not compromise the respondents' privacy. Checking the compliance of the signing party T is possible as well, since any

signed poll parameter set can later be used to prove misbehaviour. For benevolent pollsters the step of getting parameters signed only creates a small, constant overhead since it only has to be done once for every poll. After that the parameters can be distributed to each respondent during the actual poll independent of the signing party.

5 Self-Enforcing Poll Parameter Limitation

We now detail a second approach that follows the notion of self-enforcement. The basic idea is to allow proofs on the fact that a pollster has conducted more than a certain number of allowed polls in a given time frame. If such a proof is successful he has to forfeit a bounty.

In order to address the first requirement, it needs to be possible to detect and prove that different polls were conducted by the same pollster. In addition, the pollster needs to put a bounty on a successful proof on him exceeding a given poll parameter limit for a specific period. Finally, the polls and a statement about some limit need to be relatable. This can be achieved by requiring pollsters to sign both poll parameter sets and according parameter limit statements with their respective public-key. We assume the existence of a PKI and require the pollster to use his certified public-key for this. To achieve fairness, both repudiation attempts by a malicious pollster and wrong accusations by respondents need to be prevented.

Since poll participants need to be able to prove that a poll parameter set was valid at a certain point of time the parameters need not only to contain an upper bound for the number of polls but also a validity time frame. Both can be chosen freely by the pollster.

However, before partaking in a poll respondents need to check several things. First, if parameters are still valid and if the to poll parameters and the statement were signed by the same entity. Moreover, if the values chosen by the pollster are acceptable. Time frames for poll parameter limits should typically be imposed on several months rather than days or even shorter periods of time. The number of polls needs to be chosen accordingly. Poll parameter limit statements are not bound to a specific poll and can thus also be supplied separately from specific polls to the respondents. If distinct commitments are made by the pollster he needs to adhere to the lowest bound committed to since each statement is valid independently of the others. Thus, a pollster cannot gain an advantage by issuing less restrictive statements over time.

When defined in this way, the according indictment process exhibits some favourable properties. First, it can happen without the involvement of the pollster. In other words, it does not matter if he refuses to partake in his own indictment. Yet, it is not possible to gain an unfair advantage against the pollster since it is assumed that only he is in possession of his secret key. Second, no sensitive data of any respondent need to be revealed during the process as it can be the case during Basic SEP's indictment phase (see Sect. 6). Respondents can indict the pollster by presenting a commitment that they claim is being violated

and an according number of signed poll parameters. Note that since a way to uniquely identify a pollster is needed for indictment, assuming a PKI does not extend the scheme's initial assumptions.

Also noteworthy is that finding respondents and respective poll parameters can be done even if the participants did not coordinate from the beginning on. This is the case if one assumes that it is in the interest of a malicious pollster to form sets of information from multiple respondents and sell or disclose them together. Every such relation introduced by a-posteriori groupings will eventually ease the indictment process.

6 Cooperative Indictment

As elaborated in Sect. 3, there exist cases in which single respondents are unable to gather enough information to recover the secret on their own, even if all of their data are leaked. This does not need to be the result of a pollster changing parameters. It can be avoided altogether, if poll parameters are chosen so that single respondents can indict the pollster or only few bounty hunters need to collaborate (see Eq. (2)). However, this usually is neither a viable nor a desirable solution since this leads to lowered poll accuracy. Thus, when analysing the fairness of Basic SEP cooperative indictment is another important aspect to look at. Golle et al. argue that respondents can either work together or help an untrusted third party such as bounty hunters to indict the pollster [2]. Yet, they do not elaborate on the respective indictment process.

From a practical perspective respondents whose data have been leaked may be reluctant to give up the final element of uncertainty that protects their privacy. Only respondents can know which of their respective submissions were indeed real responses and which were baits. So, when asked to reveal to some other party which submissions are indeed baits they give up this security. In addition, the issue of determining the trustworthiness of the communication partner can lead to further issues. If a malicious pollster can pose as a bounty hunter or as a respondent and use the indictment process to get respondents to reveal baits the scheme's fairness cannot be guaranteed. Worse yet, a pollster can inject fake profiles in the data collection of which he knows that they are to be ignored. Yet, when looking for collaborative indictment partners he will have the ability to hamper the indictment process by for instance revealing wrong baits. Thus, the issue of identity becomes an important aspect for the scheme's fairness again.

7 Conclusions and Future Work

In this paper we have treated the scheme for data collection with self-enforcing privacy with no release of data. The effects of the until now undefined processes of parameter publication and cooperative indictment on the scheme's fairness were discussed. A way for the pollster to gain an unfair advantage by using specially constructed secrets was identified and according checks were proposed. Two approaches for fair poll parameter publication were discussed — one of them

preventive and one inspired by the concept of self-enforcement. Requirements and benefits of the respective approaches were examined. The discussed results' applicability is not limited to the analysed scheme but applies to the remaining schemes by Golle et al. as well. However, for the initial security analysis we limited our scope to the above mentioned scheme. Thus, an in depth analysis of the remaining schemes remains an open issue.

There exist questionnaire design techniques that intend to increase the confidence in the responses' correctness by, e.g., reducing the effect of acquiescence response bias. It remains to be investigated how such methods can affect Basic SEP's fairness. Last but not least, digital cash could be used for the construction of the secret. This would allow respondents to claim a bounty without involvement of the pollster. However, several details need to be investigated to see how a viable solution can be constructed.

Acknowledgements. We would like to thank M. Eian for pointing out that specific secrets can be detected efficiently and Stig F. Mjølsnes for suggesting to investigate the use of digital cash.

References

1. Golle, P., McSherry, F., Mironov, I.: Data collection with self-enforcing privacy. In: CCS '06: Proceedings of the 13th ACM Conference on Computer and Communications Security, New York, NY, USA, ACM (2006) 69–78
2. Golle, P., McSherry, F., Mironov, I.: Data collection with self-enforcing privacy. ACM Transactions on Information System Security (TISSEC) **12**(2) (2008) 1–24
3. Cox, I.J., Miller, M.L.: The First 50 Years of Electronic Watermarking. EURASIP Journal on Applied Signal Processing **2002**(2) (2002) 126–132
4. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Proceedings of CRYPTO '84 on Advances in Cryptology, New York, NY, USA, Springer Verlag (1985) 10–18
5. Camenisch, J., Stadler, M.: Proof Systems for General Statements about Discrete Logarithms. Technical Report 260, Department of Computer Science, ETH Zürich (2001)
6. Chaum, D., Pedersen, T.P.: Wallet Databases with Observers. In: CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, London, UK, Springer Verlag (1993) 89–105
7. Bella, G., Librizzi, F., Riccobene, S.: Realistic threats to self-enforcing privacy. In: IAS '08: Proceedings of the 4th International Conference on Information Assurance and Security, Washington, DC, USA, IEEE Computer Society (2008) 155–160
8. Pollard, J.M.: Monte Carlo methods for index computation (mod p). Mathematics of Computation **32**(143) (1978) 918–924