Data Collection with Self-Enforcing Privacy

PHILIPPE GOLLE
Palo Alto Research Center
and
FRANK McSHERRY and ILYA MIRONOV
Microsoft Research

Consider a pollster who wishes to collect private, sensitive data from a number of distrustful individuals. How might the pollster convince the respondents that it is trustworthy? Alternately, what mechanism could the respondents insist upon to ensure that mismanagement of their data is detectable and publicly demonstrable?

We detail this problem, and provide simple data submission protocols with the properties that a) leakage of private data by the pollster results in evidence of the transgression and b) the evidence cannot be fabricated without breaking cryptographic assumptions. With such guarantees, a responsible pollster could post a "privacy-bond," forfeited to anyone who can provide evidence of leakage. The respondents are assured that appropriate penalties are applied to a leaky pollster, while the protection from spurious indictment ensures that any honest pollster has no disincentive to participate in such a scheme.

Categories and Subject Descriptors: E.3 [**Data**]: Data Encryption; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms: Security

Additional Key Words and Phrases: privacy, data collection

ACM Reference Format:
Golle, P., McSherry, F., and Mironov, I. 2008. Data collection with self-enforcing privacy. ACM Trans. Inf. Syst. Secur. 12, 2, Article 9 (December 2008), 24 pages. DOI = 10.1145/1455518.1455521. http://doi.acm.org/10.1145/1455518.1455521.

1. INTRODUCTION

We study the problem of a pollster who wishes to collect private information from individuals of a population. Such information can have substantial value

A preliminary version of this work appeared in Golle et al. [2006].

Authors' emails: P. Golle, email: pgolle@parc.com; F. McSherry, email: mcsherry@microsoft.com; and I. Mironov, email: mironov@microsoft.com.

to the pollster, but the pollster is faced with the problem that participation levels and accuracy of responses drop as the subject matter becomes increasingly sensitive. Individuals are, understandably, unwilling to provide accurate sensitive data to an untrustworthy pollster who is unable to make concrete privacy assurances.

The same problem affects individuals who are compelled to provide sensitive data to an untrusted party. Examples such as the census and medical data highlight cases where individuals are compelled to accuracy, either through law or the threat of poor treatment, but the absence of "privacy oversight" leaves many uncomfortable. What mechanisms can be used to assure individuals that poor privacy discipline can be caught and publicly demonstrated?

We stress that this problem is different from the question of how the pollster or data collector can manage data to preserve privacy. Privacy preserving data mining research has blossomed of late and gives many satisfying answers to this question [Agrawal and Srikant 2000]. Instead, the problem we consider is that individuals may not trust the pollster to apply quality privacy protection, either because the pollster has poor privacy discipline, poor security, or simply because it is selling data on the side. Published research on privacy preserving data mining demonstrates techniques for use by a benevolent pollster, but gives no assurances to individuals who are not convinced of the benevolence of the pollster.

The focus of this article is a mechanism for submitting data to an untrustworthy pollster, such that a) leakage of private data can be caught and publicly demonstrated, and b) if private data are not leaked, the probability of presenting evidence of a leak is arbitrarily small. We stress that both of these properties are critical; the individuals must be protected from a bad pollster as much as the pollster must be protected from fraudulent accusations.

We make a distinction between individual data, and aggregate data (for example, a noisy average of respondents' data). Our schemes ensure that leakage of individual data by the pollster is detected (and punished). But some of our schemes allow the pollster to publish aggregated data provided it does not enable the inference of individual data. This is not a limitation of our schemes, but rather a useful feature, since publication of nonidentifying aggregated data is typically permitted and useful. Formal definitions of individual and aggregate data are given in Section 2.

1.1 Overview of Existing Solutions

Much research has gone into the design of data analysis mechanisms that attempt to minimize the amount of sensitive information leaked. However compelling these solutions may be, their value is greatly diminished in the absence of any guarantee that they are being applied properly. They do give substantial value when the pollster is trusted, for example, when the pollster and individuals from whom data are collected belong to the same organization, or when the pollster has legal rights to the data of the individuals.

There are several techniques to address the problem of an untrustworthy pollster, with varying features and drawbacks. Randomized response [Warner

1965; Ambainis et al. 2004] is a method in which respondents presanitize their own data by randomly altering it before submission. For example, when asked to reveal their gender, an individual could flip a coin with bias $p < 1/2$ and alter her response if the coin comes up heads. So long as the parameters of the presanitization (p, and any additional details of the presanitization process) are understood, many analyses accommodate this sort of perturbation. However, the noise levels introduced can be quite substantial. In some contexts, such as medical histories, the introduction of noise is simply a nonoption; a peanut allergy, for example, must always be reported truthfully.

Another approach is the use of trusted third parties, and their emulation through secure function evaluation [Yao 1982; Goldreich et al. 1987]. In this case, the data are collected by a trusted third party, and the untrusted pollster is only permitted to ask the trusted party certain questions of the data. The drawback of this approach is that the existence of a trusted third party is a substantial assumption, and the computational overhead involved in removing this assumption through secure function evaluation can be significant.

A third approach is to anonymize the data before submission, so that one cannot correlate sensitive features with individual identities. Mix networks [Chaum 1981; Ogata et al. 1997] allow respondents to submit data to the pollster anonymously. Unfortunately, anonymity is not feasible in many practical contexts. Mix networks can only be used to submit data that do not contain personally identifiable information (PII), so that the data themselves do not disclose information about the identity of the submitter. Whether a particular datum serves as PII depends entirely on the context, and it is rarely safe to assume that a particular parcel of data will not be disclosive when presented publicly.

In addition, or as an alternative to the deployment of privacy-preserving techniques, one may consider methods of detecting or discouraging leaks of sensitive information. This self-enforcement approach has been explored in the literature, mostly in the context of digital rights management [Boldyreva and Jakobsson 2003; Chor et al. 2000; Dwork et al. 1996; Jakobsson et al. 2002; Margolin et al. 2004]. The cryptographic schemes proposed in these articles deter a user, or a coalition of users, from sharing access to digital content by making such behavior traceable or by conditioning shared access to content on sharing some sensitive data, such as credit card numbers.

Finally, one might draw a comparison between our work and the process of tainting data, wherein submitters introduce an identifiable tracer into their submissions. One primitive example would be to encode a nonce into the least significant bits of a submission. Should the submitter see this tracer attached to their data again, they are assured that the information must have originated from the pollster. However convincing such a scheme might be to the individual, who may now severe communications with the pollster, it does little to convince the public that the pollster has done anything wrong. A public demonstration of the tainted data only confirms that either the pollster or the individual leaked the data, and does not preclude the possibility that the individual is setting up the pollster.

Existing schemes to watermark or fingerprint data [Boneh and Shaw 1998; Agrawal et al. 2003], including a publicly verifiable scheme [Pfitzmann and Schunter 1996], are designed for a setting where one data-holder manages access to its information, which is typically some large relational database or a digital movie. These techniques are not applicable in a distributed scenario, where the data are contributed by many individual participants.

1.2 Overview of Our Techniques

At the heart of our approach is the assumed presence of opportunistic third parties that we will call the *bounty hunters*, who listen for leaks of private information and assemble a case against the pollster. The bounty hunters participate in the data collection, pretending to be simple respondents (in fact, they may be). However, rather than following the cryptographic protocol for data submission, they submit "baits," whose decrypted contents provably cannot be determined without access to a secret held by the pollster. A bounty hunter herself does not know the contents of the data she submits. Since the pollster is the only individual capable of decrypting and examining the submitted bit, any report of the actual data in this message must come from the pollster, and thereby incriminates the pollster of leaking private data. Collaboration between bounty hunters is allowed, but not necessary. A single bounty hunter can produce evidence that incriminates a dishonest pollster who leaks private data.

The technical details we must discuss are the data submission process that allows respondents to submit data to the pollster, and the indictment process, in which a case is made by one or several bounty hunters against a pollster who leaked private data. There are several desirable properties of the indictment process, foremost that leakage of private data, even probabilistically, results in a viable case and that non-leakage cannot result in a viable case with high probability. These details are examined in Section 5.

1.3 Article Outline

We begin in Section 2 with a discussion of the model and several preliminary definitions and assumptions that will form the basis of our approach. Moreover, we detail several cryptographic primitives and the properties we take advantage of. In Section 3 we describe a simple approach for the case where the pollster uses the data collected from respondents only for internal consumption, and need not be able to publish any information about it (not even sanitized nonidentifying information). Section 4 outlines a submission protocol based on randomized response, which adds the property that every submission serves as bait but introduces some uncertainty into the submitted data. Sections 5 and 6 describe approaches that allow submission of precise data, but introduce the need for an interactive indictment process. Schemes from Sections 4, 5, and 6 permit limited public disclosure of analysis of the pollster's data if the pollster follows specific sanitization policies. Section 7 compares the privacy properties of our schemes for various parameter settings. Finally,

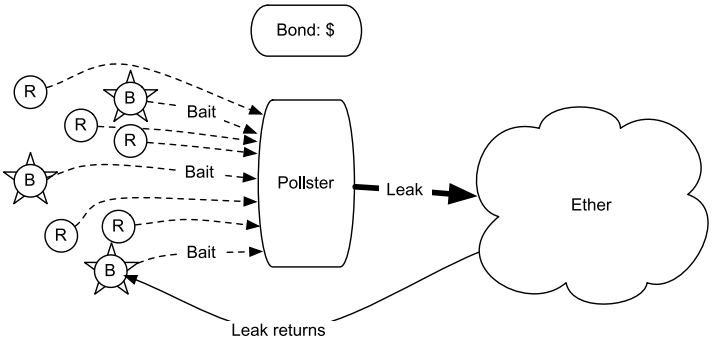

Fig. 1. The pollster collects data from respondents. A number of bounty hunters, hidden among the respondents, submit baits. The pollster cannot distinguish baits from the data submitted by respondents. A privacy breach, or leak, occurs if the pollster releases private data. Baits allow bounty hunters to offer publicly verifiable evidence of a privacy breach.

in Section 8 we conclude with a summary of the results, as well as promising directions for future investigation.

2. MODEL

We start by introducing some terminology and describing the players in our data collection processes (see Figure 1). First, there is a *pollster*, who is interested in collecting bits from a large collection of *respondents*. The pollster may also publish aggregated poll results, as long as doing so does not compromise the privacy of any respondent.

The respondents have a vested interest in the privacy of their bits, and are assumed interested in participating in a protocol that enforces privacy. To this end, the pollster offers some form of *bounty*, which it must forfeit if a privacy violation is uncovered. The bounty could be explicit in the form of a bond, or implicit in the form of penalties imposed if privacy is violated.

Lurking among the respondents are some number of bounty hunters, who masquerade as one or more respondents and attempt to ensnare the pollster in a privacy violation. The bounty hunters submit baits, which the pollster cannot distinguish from legitimate data, and hope to learn from the pollster specific information about their baits that will constitute evidence of a privacy violation. If a bounty hunter uncovers a privacy violation, this evidence can be presented to claim the bounty. We say that our scheme offers self-enforcing privacy, since it is in the best interest of the pollster to preserve the privacy of the data collected from respondents.

2.1 Defining Privacy

A self-enforcing data collection scheme ensures that a pollster who publishes sensitive data must forfeit a bounty. At the same time, the data collection scheme would ideally allow the pollster to publish aggregated poll results, as long as these results do not compromise the privacy of any respondent. Unfortunately, we do not know how to define and enforce these properties in a

strictly complementary way, that is, in such a way that any publication of the pollster is classified as either safe or helping the bounty hunter. Instead, we introduce two notions of privacy:

2.1.1 *Privacy breach.* A *privacy breach*, formally defined below along the lines of a *classical compromise* [Kenthapadi et al. 2005] is a clear violation of privacy. It amounts to the pollster releasing information that makes it possible to guess the sensitive bits confided by the respondents with a success probability non-negligibly greater than 1/2 without using any auxiliary information. A privacy breach can be thought of as a lower-bound on the privacy that the pollster must offer the respondents. We will prove that our schemes ensure that a privacy breach with certain parameters allows a bounty hunter to claim the bounty.

2.1.2 *Differential privacy [Dwork 2006].* Differential privacy is a quantifiable definition of privacy-preserving functionality. We will show that our schemes ensure that a pollster who preserves differential privacy for some range of parameters cannot lose his bounty.

As noted above, there exists a "gap" between a privacy breach and differential privacy. In other words, the pollster may release data that violate differential privacy, but do not lead to a privacy breach.

2.2 Privacy Breach

It is important to formally describe what we mean by a breach of privacy, so that we can argue that we protect against such breaches. One appealing definition is that the pollster should not release specific information about respondents to other entities. However, such breaches will not generally be detectable, as the pollster could easily release the sensitive information to parties that will not themselves pass on the information, and the breach will not be detectable without their help. Instead, we will focus on breaches that are detectable, that is, breaches for which the information released by the pollster finds its way back to the individuals who submitted that information, or agents acting on their behalf. Our focus on detectable breaches is justified by our assumption that the bounty hunters play an active role in monitoring the pollster and looking for data leaks. Naturally, the pollster should not be able to tell the bounty hunters from other agents interested in obtaining the pollster's data.

It will be critical that the respondents are able to identify a privacy breach as such. One example would be seeing one's private data made available, though less direct observations, such as for example being contacted on one's cell phone by a solicitor, can lead to similar conclusions.

Formally, we consider a model where the data received by the pollster are encoded as an n-bit vector $v = \{v_1, \ldots, v_n\}$. Note that the values received by the pollster may not be known by the respondents.

Definition 2.1. A (ℓ, ϵ)-*privacy breach* exists when ℓ indices $i_1, \ldots, i_\ell$ are identified such that any assignment of $v_{i_1}, \ldots, v_{i_\ell}$ consistent with the

information published by the pollster agrees with v on at least $1/2 + \epsilon$-fraction of the entries.

2.3 Differential Privacy

A natural question that arises in the presence of a posted privacy bond is whether the pollster can analyze and release any properties of the private data collected from respondents. Might it be that all useful functions reveal too much about the structure of baits so that the bond must be forfeited as soon as the pollster publishes any information at all about the data collected?

In this section, we introduce ϵ-differential privacy, a natural definition of privacy proposed in Dwork et al. [2006] and Dwork [2006]. We argue that the pollster can publish the results of any analysis that preserves differential privacy, without incurring a substantial risk of having to forfeit the bounty. Indeed, the chance of producing evidence of a privacy breach against a pollster is exponentially small if the pollster releases only information that preserves differential privacy.

Definition 2.2. A randomized function f over data sets gives *ϵ-differential privacy* if for any two data sets X_1 and X_2, which differ in at most one point, and $S \subseteq \mathit{Range}(f)$,

$$\Pr[f(X_1) \in S] \quad \leq \quad \exp(\epsilon) \times \Pr[f(X_2) \in S].$$

In our application, the output of the function f is the information that the pollster releases about the data. The definition of differential privacy ensures that this information is not substantially affected by a respondent's presence in (or absence from) the data. Intuitively, if all the information published by the pollster preserves differential privacy, a bounty hunter cannot learn the data of any respondent and thus can also not learn information about any bait. We will use this property to show that publication of the results of analyses that preserve ϵ-differential privacy do not give bounty hunters enough information about baits to successfully claim the bounty with non-negligible probability.

Differential privacy is discussed in more detail in Dwork et al. [2006] and Dwork [2006], in which methods are presented for performing several common data analyses in a way that preserves differential privacy. Examples include histogram computations such as OLAP, as well as more algorithmic analyses such as Principal Components Analysis, k-means clustering, perceptron classification, and ID3 decision trees construction.

We stress that our data collection schemes are not bound to ϵ-differential privacy. This definition of privacy was chosen only to demonstrate that the privacy of our mechanisms can coexist with nontrivial data analyses. Differential privacy is among the stronger definitions of privacy, and is therefore easier to accommodate. Differential privacy is applied in Sections 4 and 5.

2.4 Cryptographic Building Blocks

The approaches we present make use of cryptography to ensure that certain information is concealed from respondents, and that other information can be

presented irrefutably. We now detail some of the cryptographic primitives that we use and their properties.

2.4.1 *Secure channels.* In our schemes, respondents will submit to the pollster data encrypted with homomorphic public-key encryption schemes, such as ElGamal or RSA. These encryption schemes naturally do not provide chosen-ciphertext security. It is thus imperative that these public-key ciphertexts be submitted over a secure channel, such as TLS. In fact, it is easy to demonstrate that the security of respondents' submissions is compromised if our schemes were used without an additional layer of (symmetric-key) encryption.

2.4.2 *ElGamal cryptosystem.* ElGamal is a randomized public-key encryption scheme. Let G be a group, and let $g \in G$ be a generator of a multiplicative subgroup G_q of order q where the Decisional Diffie-Hellman problem is hard. The secret key is an element x chosen at random from Z_q. The corresponding public key is the value $y = g^x$. The encryption of a plaintext $m \in G_q$ is a pair (g^r, my^r) for a value r chosen at random in Z_q. To decrypt a ciphertext (A, M), the value $m = M/A^x$ is computed. We will use two important properties of ElGamal:

—**Multiplicative homomorphism:** Consider two ElGamal ciphertexts $C_1 = (g^r, m_1y^r)$ and $C_2 = (g^s, m_2y^s)$ for plaintexts m_1 and m_2. The component-wise product $C_1.C_2 = (g^{r+s}, m_1m_2y^{r+s})$ is an ElGamal ciphertext for m_1m_2.

—**Re-encryption.** Let (g^r, my^r) denote an encryption of a plaintext m. Let s be a random value in $\mathbb{Z}_q$. The pair (g^{r+s}, my^{r+s}) is also an encryption of m. The new pair is called a re-encryption of the first ciphertext. Note that a ciphertext can be re-encrypted without knowledge of m or of the secret key x.

2.4.3 *Proof of plaintext knowledge (KPT).* Let $E(m) = (g^r, my^r)$ be an encryption generated by a prover. The prover can prove to a verifier that she knows the plaintext m by proving that she knows $\log_g(g^r)$. This can be done with a protocol by Schnorr [Schnorr 1991]. The protocol can be made noninteractive with the Fiat-Shamir heuristic. We denote an instance of this protocol for an ElGamal ciphertext C as $KPT(C)$.

2.4.4 *Proof of correct decryption (PCD) [Chaum and Pedersen 1993].* A prover proves to an honest verifier that an ElGamal ciphertext (C, M) decrypts to a plaintext m. The proof consists of showing that $\log_g(y) = \log_C(M/m) = x$ without leaking any information about the secret key x. We denote an instance of this protocol to prove correct decryption of an ElGamal ciphertext C as $PCD(C)$.

2.4.5 *Proof of correct re-encryption (PCR) [Chaum and Pedersen 1993].* A prover proves to an honest verifier that an ElGamal ciphertext (g^s, my^s) is a re-encryption of a ciphertext (g^r, my^r) without leaking any other information. The proof consists of showing that $\log_g(g^s/g^r) = \log_y((my^s)/(my^r)) = s - r$, without leaking any information about the value $s - r$. The computational cost of this protocol is two modular exponentiations for the prover and four modular

exponentiations for the verifier. We denote an instance of this protocol to prove that an ElGamal ciphertext C_2 is a re-encryption of C_1 as $PCR(C_1 \rightsquigarrow C_2)$.

2.4.6 *Discrete logarithm proof systems [Camenisch and Stadler 1997].* An efficient zero-knowledge proof can be constructed for any monotone boolean formula whose atoms consist of the protocols to prove plaintext knowledge (KPT), correct decryption (PCD), or correct re-encryption (PCR).

2.4.7 *Verifiable mixing [Groth 2002; Neff 2001].* Let $L = \{(A_i, M_i)\}$ and $L' = \{(A'_i, M'_i)\}$ be two lists of ElGamal ciphertexts. A verifiable mixing protocol allows a prover to prove to an honest verifier the existence of a permutation π and a sequence of exponents γ_j such that $(A'_j, M'_j) = (A_{\pi(j)}g^{\gamma_j}, M_{\pi(j)}y^{\gamma_j})$, without leaking any information about π or the values γ_j. Given n input ciphertexts, the computational cost of the most efficient verifiable mixing protocol [Groth 2002] is $6n$ modular exponentiations for the prover and $6n$ modular exponentiations for the verifier.

3. SELF-ENFORCING PRIVACY WITH NO RELEASE OF DATA

In this section, we present a scheme that allows the pollster to collect data from respondents, but not to release any information about the data collected.

The scheme is structured as follows. The pollster commits to a secret binary string by publishing encryptions of the bits of the secret string under a randomized public-key encryption scheme, such as ElGamal, which is homomorphic and allows for re-encryption of ciphertexts. Each time a respondent submits a bit, she has a choice of either submitting an encrypted bit of her own data or preparing a bait by re-encrypting any of the pollster's secret bits. The pollster decrypts all the ciphertexts received and thus recovers the data submitted by respondents. Since the pollster cannot distinguish baits from regular submissions, some baits will unavoidably be decrypted if the pollster leaks a substantial fraction of the data. Decrypted baits reveal some of the bits of the pollster's secret string. Once enough of the secret bits are known to the injured parties, they can claim the bounty by proving knowledge of the secret string.

In this section and throughout the article, we assume that respondents are labelled with unique identifiers $P_1, \ldots, P_n$.

3.1 Setup

The pollster outputs public parameters for a public-key encryption scheme E that is semantically secure under re-encryption and has a multiplicative homomorphism. In what follows, we use ElGamal. The 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-Hellman problem is hard.

3.2 Commitment to the Bounty

Let k be a security parameter (e.g., $k = 160$). The pollster chooses a k-bit secret value $\beta = b_1 \ldots b_k$. The pollster outputs $E(g^{b_i})$ for $i = 1, \ldots, k$ and proves that

these ciphertexts are well-formed by showing that each ciphertext decrypts either to g^0 or g^1. This is done with a (disjunctive) discrete logarithm proof system consisting of two proofs of correct decryption (see Section 2.4). Using the multiplicative homomorphism of E, the pollster computes $\prod_{i=1}^{k} E(g^{b_i})^{2^i} = E(g^{\beta})$. The pollster then decrypts this value, proves correct decryption with the protocol $PCD(E(g^{\beta}))$ described in Section 2.4, and outputs the commitment g^{β}. A bounty is then offered to anyone who recovers the secret value β.

3.3 Data Submission

In the data submission step, a respondent sends to the pollster either one true bit of data or a bait.

3.3.1 *Sending one true bit of data.* To send a bit $b \in \{0, 1\}$ to the pollster, a respondent P_i computes the randomized ciphertext $E(g^b)$ and sends the resulting value to the pollster over a secure channel (e.g., using TLS).

3.3.2 *Sending a bait.* To send a bait to the pollster, the respondent chooses a random index $r \in \{1, \ldots, k\}$, re-encrypts the ciphertext $E(g^{b_r})$ and sends the re-encrypted ciphertext to the pollster over a secure channel.

3.4 Data Collection

The pollster receives ElGamal ciphertexts from respondents. Since ElGamal is semantically secure under re-encryption, the pollster cannot distinguish true bits from baits. The pollster then decrypts all ciphertexts $C = E(g^{b_i})$ and recovers the corresponding plaintexts. Only well-formed plaintexts (i.e., those that decrypt to g^0 or g^1) are tallied. Malformed plaintexts are discarded.

3.5 Claiming the Bounty

3.5.1 *Honest pollster.* This scheme does not allow the pollster to publish anything about the data collected. We show first that corrupt respondents cannot fraudulently claim the bounty of an innocent pollster. If the pollster leaks no information about data collected from respondents, claiming the bounty is equivalent to recovering the value β from the commitment g^{β}. Since the discrete logarithm problem is assumed hard in the group G generated by g, this problem is computationally intractable. Thus corrupt respondents cannot wrongly claim the bounty of an innocent pollster.

3.5.2 *Dishonest pollster.* We consider next a dishonest pollster, and show that the bounty can be recovered if the pollster publishes data that result in a privacy breach. Let us start with a simple example. If the pollster leaks $\ell < k$ baits, respondents can recover the secret β in time $2^{(k-\ell)/2}$ using the technique of Pollard [1978] and present β as evidence of the pollster's misbehavior to claim the bounty. Note that the verification process is noninteractive: the correctness of β is verified against the commitment g^{β}, without communicating with the pollster. The correctness of the bounty is also publicly verifiable without the involvement of the pollster.

More generally, let us consider a pollster who publishes data that result in a privacy breach. For example, the pollster may leak the data collected from respondents with noise added. The following proposition shows that a privacy breach allows bounty hunters to recover all the bits of the pollster's secret with high probability.

PROPOSITION 3.1. *Consider a pollster who commits a* (ℓ, ϵ)*-privacy breach. Recall that* k *denotes the size of the pollster's secret. Let* $0 < \alpha < 1$ *denote the fraction of baits among the bits submitted by respondents and bounty hunters. If* $\ell > k/(\alpha\epsilon^2)$*, the bounty hunters can (with high probability) reconstruct the secret* β *with no computational effort.*

PROOF. Let us denote the data received by the pollster as an n-bit vector $v = \{v_1, \ldots, v_n\}$. By definition, an (ℓ, ϵ)-privacy breach means that a set of ℓ indices $i_1, \ldots, i_\ell$ is identified such that any assignment of $v_{i_1}, \ldots, v_{i_\ell}$ consistent with the information published by the pollster agrees with v on at least $1/2+\epsilon$-fraction of the entries.

Among the values $v_{i_1}, \ldots, v_{i_\ell}$, the number of baits is $\alpha\ell$. Now let us consider a bit b_i of the pollster's secret β. The number of baits in $v_{i_1}, \ldots, v_{i_\ell}$ that are re-encryptions of the bit b_i is $\alpha\ell/k$. By definition of a privacy breach, each of these baits is correct with probability greater than $1/2 + \epsilon$. If a majority of these $\alpha\ell/k$ values are 0, we conclude that $b_i = 0$ (and otherwise $b_i = 1$).

Let X be a random variable defined by the sum of the $\alpha\ell/k$ baits that are re-encryptions of the bit b_i. According to the Chernoff bound,

$$\Pr[X < \alpha\ell/(2k)] < e^{-(\alpha\ell/k)(1/2+\epsilon)(1-1/(1+2\epsilon))^2/2}.$$

The probability of error is thus small if $\ell = O(k/(\alpha\epsilon^2))$. This concludes the proof. □

Let us consider a numerical example. If the pollster commits to a 160-bit secret ($k = 160$) and leaks correct bits with probability $1/2 + \epsilon$, where $\epsilon = 1/4$, and if the respondents submit a bait with probability $\alpha = 10\%$, then $12,800$ bits are required to recover β with modest computational effort (2^{40} modular exponentiations).

We stress that this scheme is secure for the pollster only if it releases no information whatsoever about the data collected. The following example illustrates the danger for the pollster of releasing even seemingly innocuous data.

Consider a pollster who intends to publish the noisy gender majority for each ZIP code in the survey. For appropriately chosen parameters of the noise, this information can be disclosed without a privacy breach. Still, the scheme described in this section does not allow for the safe release of this information.

Indeed, an unscrupulous bounty hunter may create sufficiently many false identities in a given ZIP code area, and let all these identities submit as baits re-encryptions of the same secret bit of the pollster's secret. The bounty hunter may succeed in biasing the results of the poll so that the noisy majority will be equal to the value of this secret bit with high probability. Repeating this attack will eventually allow the bounty hunter to learn all the bits of the pollster's secret and claim the bounty.

In the rest of this article, we propose improved schemes that will allow the pollster to safely release sanitized nonidentifying information about the data collected.

4. SELF-ENFORCING PRIVACY AND RANDOMIZED RESPONSE

The scheme described in the previous section requires different processes for submitting true answers and baits. It calls for proactive bounty hunters, who may have an incentive to create multiple fake identities that crowd out real contributors and compromise the poll's validity.

In this section, we propose a different scheme based on the concept of randomized response, where each response is a bait and the role of bounty hunters in the survey is strictly passive.

4.1 Basic Scheme with Randomized Response

4.1.1 *Setup.* The pollster outputs public parameters for an ElGamal encryption scheme denoted E. As in Section 3, we denote $g \in G$ the generator of a multiplicative subgroup G_q of order q in which the Decisional Diffie-Hellman problem is hard.

4.1.2 *Commitment.* The scheme is parameterized with κ, the order of the group element g in G (κ may be 160 in most scenarios). The pollster chooses κ bits $b_1, \ldots, b_\kappa$ at random, such that exactly half of them be ones. Let β denote the integer whose binary representation is $b_1, \ldots, b_\kappa$. The pollster outputs $E(g^{b_i})$ for $i = 1, \ldots, \kappa$. The pollster proves that the ciphertexts are well-formed, that is, are encryptions of either g^0 or g^1. Next, using the multiplicative homomorphism of E, the pollster computes $E(g^\beta) = \prod_{i=1}^{\kappa} E(g^{b_i})^{2^i}$. The pollster then provably decrypts this value and outputs g^β. The bounty is placed on the value β. Finally, the pollster produces a list of $k - \kappa$ ciphertexts, which are encryptions of g^0. The combined list of k ciphertext plays the same role as in Section 3.

4.1.3 *Data submission.* Let b denote the bit to be submitted by a respondent. The respondent chooses a random index $i \in \{1, \ldots, k\}$. If $b = 0$, the respondent sends to the pollster a re-encryption of the ciphertext $E(g^{b_i})$. If $b = 1$, the respondent uses the multiplicative homomorphism of ElGamal to compute the ciphertext $E(g^{1-b_i}) = E(g)/E(g^{b_i})$ and sends this ciphertext to the pollster over a secure channel. Let C denote the ElGamal ciphertext sent to the pollster.

The respondent must also submit a proof of correct operation. The respondent gives a proof to the pollster of the following discrete-log system (see Section 2.4):

$$\Big(\bigvee_{i=1}^{k} PCR\Big(E(g^{b_i}) \rightsquigarrow C\Big)\Big) \vee \Big(\bigvee_{i=1}^{k} PCR\Big(E(g)/E(g^{b_i}) \rightsquigarrow C\Big)\Big).$$

According to Camenisch and Stadler [1997], the cost of this proof is $6k - 1$ modular exponentiations for the prover (the respondent) and $6k$ modular exponentiations for the verifier (the pollster). The purpose of this proof it to prevent respondents from cheating by submitting nonrandomized replies that would carry more weight than randomized ones.

The probability that a respondent's bit is inverted, that is, the randomization parameter of the randomized response scheme, is $p = \kappa/(2k)$.

4.1.4 *Data collection and claiming the bounty*. These steps are exactly as in Section 3.

The probability that the bond is claimed is provably reducible to the discrete logarithm problem of recovering β from g and g^β in the group G. The only constraint is that the number of nonzero bits in β is exactly half its length, which reduces the complexity of the problem by a factor less than κ.

It is natural to compare this approach with the simpler randomized response schemes described in the introduction, in which the respondents prerandomize their own data. The values submitted in our scheme have no greater statistical fidelity than in the simpler scheme. The important distinction is that, in our scheme, the interests of the pollster are aligned with the privacy concerns of the participants: a value p close to zero gives very accurate answers but puts the bounty at risk. The privacy of the individuals is not a result of choosing p close to 1/2, as in classical randomized response, but inherent for all values of p.

4.2 Variant that Allows Release of Some Data

The data collected from respondents are most useful when the pollster is able to analyze it and can act on the analyses (or even publish the results of the analyses) without fear of forfeiting the bounty (as long as the results of the analyses do not compromise the privacy of respondents). While the scheme of Section 4.1 ensures that privacy breaches are punished, the pollster would also like assurances about what sort of behavior (or publication) is allowed, based on the data collected. If the publication of certain data is allowed, because it poses no threat to the privacy of respondents, the publication of that data should not allow a bounty hunter to successfully claim the bounty.

When the pollster performs queries over the bits submitted by the respondents, it is in fact performing queries over bits of its own secret. Publishing the results of such queries raises the concern that the pollster may accidentally reveal information about its secret bits. The pollster would like to restrict itself to queries that guarantee the "privacy" of its own secret, so that it runs no risk of having to forfeit the bounty. The property desired by the pollster is the same as ϵ-differential privacy for respondent data: the distribution over results should not be substantially affected by the modification of one of the pollster's secret bits.

To achieve this property, we propose a simple variant of the data submission protocol of Section 4.1. Recall from Section 4.1 that the pollster outputs k ciphertexts $E(g^{b_i})$ for $i = 1, \ldots, k$ in the commitment step. Intuitively, the goal of the variant presented here is to prevent one respondent (or a set of colluding

respondents) from all submitting the *same* ciphertext $E(g^{b_i})$. We achieve this with the following data submission protocol:

(1) The pollster re-encrypts the ciphertexts $E(g^{b_i})$ for $i = 1, \ldots, k$ and permutes them according to a permutation π chosen uniformly at random and known only to the pollster. The pollster outputs the permuted set $E(g^{b_{\pi(i)}})$ for $i = 1, \ldots, k$.
(2) Let b denote the bit to be submitted by a respondent. The respondent chooses a random index $j \in \{1, \ldots, k\}$. Let i denote the value (not known to the respondent) such that $j = \pi(i)$. If $b = 0$, the respondent computes a re-encryption of the ciphertext $E(g^{b_{\pi(i)}})$. If $b = 1$, the respondent uses the multiplicative homomorphism of ElGamal to compute the ciphertext $E(g^{1-b_{\pi(i)}}) = E(g)/E(g^{b_{\pi(i)}})$. Either way, let C denote the ciphertext computed by the respondent. The respondent sends the pollster a commitment to C.
(3) The pollster reveals the permutation π and proves correct mixing in step 1 (see Section 2.4 for details on how that is done). If the verification fails, the respondent aborts the data submission process.
(4) The respondent outputs C, together with a proof of a discrete-log system that shows that C is either a re-encryption of $E(g^{b_{\pi(i)}})$ or of $E(g)/E(g^{b_{\pi(i)}})$, as in Section 4.1.
(5) The pollster checks C against the commitment received in step 2, and checks the discrete-log proof system. If both are correct, the bit from the respondent is accepted.

A malicious respondent may attempt to skew the distribution of the indices $\pi(i)$ by not completing step 4. To ensure a near-uniform distribution (with statistical distance from the uniform less than $1/k$), the pollster should use a random index if the submission protocol is aborted after the permutation π is revealed.

Now consider ϵ-differential privacy as applied to the respondent data. If the information released by the pollster preserves ϵ-differential privacy for the respondents, then the distribution over its outputs does not change substantially (as a function of ϵ) if any respondent changes its submitted value. Let s_i denote the number of respondents from a query set S whose submission is a re-encryption of bit b_i. Since a change in the value of the secret bit b_i results in a change of at most s_i values, any computation that preserves ϵ-differential privacy for the respondents' data also preserves (ϵs_i)-differential privacy for bit b_i of the pollster's secret.

THEOREM 4.1. *An ϵ-differential privacy query over the set S increases the probability of the bounty being claimed by at most* $\exp(\epsilon\kappa \max_i s_i)$.

PROOF. Consider the probability that the bounty hunter succeeds in identifying the $\kappa/2$ secret locations i for which $b_i = 1$, taken first over the randomness in the selection of the locations, and then over the randomness given by ϵ-differential privacy. Take $c = \kappa \max_i s_i$ as the largest number of respondents whose received data would change as a result of an arbitrary change in the $\kappa/2$ locations of non-zero bits. The bounty hunter's distribution over guesses

is conditioned on the locations chosen, but differential privacy guarantees that no guess increases in probability by more than a factor of $\exp(\epsilon c)$. We can therefore remove the bounty hunter's dependence on the actual location at the cost of a factor of $\exp(\epsilon c)$.

We have

$$\begin{aligned} & \Pr_{\text{location}} \Pr_{\text{guess}}[\text{guess} = \text{location} \mid \text{location}] \\ \leq & \Pr_{\text{location}} \Pr_{\text{guess}}[\text{guess} = \text{location}] \exp(\epsilon c) \\ = & \exp(\epsilon c) / \binom{\kappa}{\kappa/2}. \end{aligned}$$

The final step follows from the observation that no matter the distribution over the guess, the uniform distribution over the actual location makes the probability one over the number of possible locations. □

If the query is independent of the distribution of respondents, $\sum_{i:b_i=1} s_i$ is unlikely to greatly exceed $(1-p)\|S\|$.

If the query is permitted to depend on the distribution, perhaps because the respondents themselves pose the questions in an attempt to trap the pollster, then $\sum_{i:b_i=1} s_i$ could be as large as $\|S\|$, but even in this case the pollster can still choose ϵ and k to yield meaningful results.

4.3 A Stronger Bound for Sum Queries

In the case where the query is independent of the assignment of respondents to bits, we can occasionally prove a stronger bound for the scheme of Section 4.2. Consider the query that counts the number of respondents from S whose bit is set. If the pollster were to change the location of one of its nonzero bits, the total sum would change by at most the difference in the sums for the two locations. If the distribution of respondents is uniform, this difference can be substantially smaller than the sums themselves, improving substantially on the bound above. The following lemma is a standard balls-and-bins argument of the number of balls in a bin tightly concentrated around its expected value. As a corollary, the lemma implies that the difference between the number of balls in two bins is likely to be small compared to the total number of the balls in both bins, which corresponds to the change in a sum-query's answer if the location of a nonzero bit changes.

LEMMA 4.2. *Letting s_i be the random variable denoting the number of respondents in bin i, with probability at least $1-\delta$, for all i we have $(s_i - \mu)^2 \leq 4(s/k)\ln(k/\delta)$, provided that $\delta > \exp(-s/k)$.*

Letting d be the change in the value of the sum above, an identical change can be attained by changing the values of d respondents. If the pollster maintains ϵ-differential privacy for the respondents, the pollster is assured of ϵd-differential privacy for the location of each of its non-zero bits, even though substantially more than d respondents may live at each location.

THEOREM 4.3. *For any counting query that is independent of the distribution of respondents to bins that maintains ϵ-differential privacy of the respondents data with probability at least $1-\delta$ the probability of a bounty being claimed is at most* $\exp(2\epsilon\kappa\sqrt{(s/k)\ln(k/\delta)})/\binom{\kappa}{\kappa/2}$*, which vanishes for large enough k.*

PROOF. We start with the observation that for each of the $\kappa/2$ possible locations for the set bits, the number of positive and negative respondents are within $c' = 2\sqrt{(s/k)\ln(k/\epsilon)}$ of their mean, with probability at least $1-\epsilon$. Conditioned on this event holding, changing the location of the $\kappa/2$ bits results in a change of at most $c = \kappa c'$ to the sum. A change of c to the sum could be caused by the alteration of as many respondents data, but ϵ-differential privacy ensures that the probability of no event should increase by more than a factor of $\exp(\epsilon c)$ due to such a change. The proof follows in a form identical to that of Theorem 4.1. □

5. A SCHEME WITH INTERACTIVE INDICTMENT BASED ON RSA

In this section, we propose another scheme that allows the pollster to release information about the data collected as long as it does not violate the privacy of any nontrivial fraction of respondents. In a nutshell, our scheme works as follows. The bounty hunter prepares encryptions of unknown bits and submit them as baits. Should the pollster leak information about these bits, the bounty hunter indicts the pollster by presenting the bits and a proof of the baits' validity in order to claim the bounty. After the indictment, the onus is on the pollster to refute the accusation, which can be achieved by proving that sufficiently many bits decrypt to different values than alleged by the bounty hunter.

5.1 Setup

Let E denote a semantically secure public-key encryption scheme (e.g., RSA in what follows) and let D denote the corresponding decryption function. The pollster outputs public parameters for E. Let h be a hash function and let f be another hash function whose image is the set of ciphertexts of E. In our proof of security, we model h and f as random oracles. In the real world the functions are instantiated based on cryptographically strong hashes, such as SHA-256.

5.2 Sending a Bit to the Pollster

To send a bit $b \in \{0, 1\}$ to the pollster, a respondent P_i chooses a value r such that the least significant bit of $h(P_i||r)$ is b. The respondent sends P_i and $E(r)$ to the pollster.

5.3 Decryption

Given a respondent identifier P and a ciphertext C, the pollster decrypts C to recover the plaintext r, then computes the least significant bit b of $h(P||r)$.

5.4 Sending a Bait to the Pollster

To send a bait to the pollster, the respondent chooses a random value s, computes $f(s)$ and sends to the pollster P_i and $f(s)$. Notice that neither the decryption of $f(s)$ nor the bit recovered by the pollster is known to the bounty hunter.

5.5 Accusing the Pollster

If the pollster releases uniquely identifiable bits, some of which can be linked to the baits, the bounty hunter can indict the pollster. For some integer parameter n_0 (whose value is discussed later), the indictment consists of $n > n_0$ *distinct* triples of the form

$$\langle P_i,\ s_i,\ b_i \rangle,$$

which we call *exhibits*. An exhibit is *valid* if and only if the bit decrypted by the pollster, i.e. the least significant bit of $h(P_i||D(f(s_i)))$ is equal to b_i.

The pollster can contest the indictment by demonstrating that at least $(1/2 - w_n)n$ of the alleged exhibits are invalid. The minimum number of exhibits n_0 and the exact form of w_n, which lies between 0 and 1/2 and serves to protect the pollster, will be discussed later. The pollster proves that an exhibit is invalid by outputting $r_i = D(f(s_i))$ and demonstrating that the least significant bit of $h(P_i||r_i)$ is not b_i.

If the pollster cannot defend herself or refuses to do so, the bounty must be forfeited. Note that this solution requires the pollster to be online for the indictment process, but it does not rely on a trusted third party.

5.6 Security

We note first that the reason for using RSA in this scheme, instead of ElGamal as in previous schemes, is to give respondents the ability to select a random valid ciphertext for which they do not know the corresponding plaintext. We note also that properly constructed baits are indistinguishable from other submissions, and encode bits that are uncorrelated and provably unknown to the bounty hunter. Next, we show that a pollster whose data disclosure policy preserves ϵ-differential privacy cannot be convicted by an over-zealous bounty hunter.

PROPOSITION 5.1. *If the data queries answered by the pollster preserve ϵ-differential privacy, the probability that any bounty hunter can claim the bounty is less than*

$$\max_{n \geq n_0} \exp(n\epsilon - nw_n^2/2).$$

PROOF. To analyze the probability of successful indictment given differential privacy access to a data set, we consider the joint probability density between the indictment made, I, (a set of locations and guesses at the values at these locations) and the data set, d, which we take to have a random prior distribution from the point of view of the attacker. Using the predicate $V(I, d)$

to indicate valid indictments I for a data set d, ie: at least a $1/2 + w_n$ fraction of the exhibits are correct, the probability of successful indictment is

$$\int_I \sum_d \Pr[I \wedge d]\ V(I,d) \;=\; \int_I \Pr[I] \sum_d \Pr[d|I]\ V(I,d)\ .$$

For each I, we now decompose d into the product of two random variables, d_I and d'_I, the restriction of the data set to locations identified by the indictment and the remaining data. Noting that $V(I,d)$ depends only on d_I, and extending the notation to $V(I,d_I)$ with the same meaning,

$$= \int_I \Pr[I] \sum_{d'_I} \sum_{d_I} \Pr[d_I \wedge d'_I|I] V(I,d_I)$$

$$= \int_I \Pr[I] \sum_{d'_I} \Pr[d'_I|I] \sum_{d_I} \Pr[d_I|I,d'_I]\ V(I,d_I)\ .$$

Apply Bayes' rule to $\Pr[d_I|I,d'_I]$ gives $\Pr[I|d_I,d'_I](\Pr[d_I|d'_I]/\Pr[I|d'_I])$, and noting that by independence $\Pr[d_I|d'_I] = \Pr[d_I]$,

$$= \int_I \Pr[I] \sum_{d'_I} \Pr[d'_I|I] \sum_{d_I} \Pr[I|d_I,d'_I] \frac{\Pr[d_I]}{\Pr[I|d'_I]}\ V(I,d_I)\ .$$

The definition of differential privacy allows us to conclude that $\Pr[I|d_I,d'_I] \leq \exp(\epsilon n)\Pr[I|d'_I]$, which after canceling with the $\Pr[I|d'_I]$ in the denominator gives

$$\leq \int_I \Pr[I] \sum_{d'_I} \Pr[d'_I|I] \sum_{d_I} \Pr[d_I] V(I,d_I) \exp(\epsilon n)\ .$$

For any fixed indictment I, the probability that d_I satisfies $V(I,d_I)$ is at most $\exp(-nw_n^2/2)$ by a Chernoff bound (since we assume that RSA is one-way and we model the hash functions f and h as random oracles), giving

$$\leq \int_I \Pr[I] \sum_{d'_I} \Pr[d'_I|I] \exp(\epsilon n) \exp(-nw_n^2/2)\ .$$

The $\exp(n\epsilon - nw_n^2/2)$ term is independent of I and d'_I, and as their densities integrate to one, the equation achieves our stated bound. □

The pollster must determine a value of ϵ that permits sufficient utility without compromising the security of the bounty. Safe values of ϵ in turn depend on the values n_0 and w_n that govern the indictment rules. These values must be chosen to permit a sufficient level of safe disclosure. At the same time, respondents should also insist on realistic settings of n_0 and w_n to ensure that bounty hunters are able to catch privacy leaks.

It is also worth noting that the probability that a bounty hunter succeeds in a fraudulent claim depends only on the number of exhibits n, and not on

the total number of baits submitted. Were this not the case, there would be a strong incentive for bounty hunters to flood the system with baits, corrupting the integrity of the poll.

6. A SCHEME WITH INTERACTIVE INDICTMENT BASED ON ELGAMAL

In this section, we present a variant of the scheme of Section 5 based on ElGamal. Like the scheme of Section 5, the scheme presented in this section allows the pollster to release information about the data collected from respondents as long as it does not violate the privacy of a non-trivial fraction of respondents. Since ElGamal ciphertexts (on elliptic curve groups) can be shorter than RSA ciphertexts, the scheme based on ElGamal offers better communication complexity than that based on RSA. On the downside, the ElGamal-based scheme of this section assumes the existence of a trusted judge who supervises the indictment process when the pollster stands accused of leaking private data. The description of the scheme follows.

6.1 Pollster Setup

The pollster outputs public parameters for an ElGamal encryption scheme denoted E. As in Section 3, we denote $g \in G$ the generator of a multiplicative subgroup G_q of order q in which the Decisional Diffie-Hellman problem is hard. Let h denote a cryptographically strong hash function, such as SHA-256.

6.2 Sending a Bit to the Pollster

Let b denote the bit to be submitted by a respondent. The respondent first contacts the pollster and indicates its intention to submit a bit. When contacted, the pollster chooses a random value r_i and sends to the respondent the ciphertext $E(r_i)$ together with the pollster's signature σ_i on $E(r_i)$. The respondent then proceeds as follows, depending on whether it intends to submit a true bit of data or a bait:

6.2.1 *Sending one true bit of data.* To send a bit $b \in \{0, 1\}$ to the pollster, the respondent chooses a random value s_i such that the least significant bit of $h(s_i)$ is b. The respondent then computes the randomized ciphertext $E(s_i)$ and sends that value to the pollster over a secure channel (e.g., using TLS).

6.2.2 *Sending a bait.* To send a bait to the pollster, the respondent chooses a random nonzero value s_i, computes the ciphertext $E(r_i \cdot s_i)$ using ElGamal's multiplicative homomorphism, and sends the resulting ciphertext to the pollster over a secure channel.

6.3 Data Collection

The pollster receives ElGamal ciphertexts from respondents. Since ElGamal is semantically secure under re-encryption, the pollster cannot distinguish true bits from baits. The pollster decrypts all ciphertexts $E(t_i)$ and recovers the least significant bit b_i of the value $h(t_i)$.

6.4 Accusing the Pollster

If the pollster releases uniquely identifiable bits, some of which can be linked to the baits, the bounty hunter can indict the pollster. The indictment consists of $n > n_0$ *distinct* triples of the form

$$\langle E(r_i),\ \sigma_i,\ s_i,\ b_i\rangle,$$

which we call *exhibits*.

6.5 Validating Exhibits

If an indictment is brought up against the pollster, the exhibits are submitted to the judge, who privately reviews their validity. An exhibit is valid if and only if the following conditions hold:

(1) The signature σ_i is a valid pollster's signature on $E(r_i)$.
(2) The least significant bit of $h(r_i \cdot s_i)$ is equal to b_i.

The judge first eliminates exhibits which do not satisfy the first property. The judge must then determine which exhibits satisfy the second property, with help from the pollster. The pollster can contest the validity of an exhibit by demonstrating that it does not satisfy the second property. To demonstrate to a trusted judge that an exhibit $\langle E(r_i),\ \sigma_i,\ s_i,\ b_i\rangle$ is invalid, the pollster decrypts the value $E(r_i)$ and outputs r_i for the judge. This allows the judge to check whether the least significant bit of $h(r_i \cdot s_i)$ is equal to b_i.

6.6 Security

The security properties of this scheme are identical to the scheme of Section 5. We note that it is important that exhibits be validated in private by a trusted judge. This prevents participants from learning which exhibits are valid, and which are invalid. The reason why participants must be prevented from learning this information is as follows. If they did learn this information, they should be prevented from ever submitting the same exhibit twice (it is trivial to create valid exhibits if two tries are allowed). But preventing multiple submissions opens the door to an attack, in which the pollster preventively accuses himself with invalid exhibits. This self-accusation does not cause the pollster to forfeit his bond, since the exhibits are invalid. But it ensures that the same exhibits can not later be used by participants to accuse the pollster, and thus allows the pollster to leak the data corresponding to these exhibits without fear of retribution.

7. EMPIRICAL MEASUREMENTS OF BOUNDS

In this section we briefly look at actual parameter settings derived from the bounds we have proven, considering a range of parameters that might be realistic, and contrasting the trade-offs of each approach. We will consider four approaches: the two analyses of Randomized Response, corresponding to Sections 4.2 and 4.3, and two instances of the scheme from Section 6, with the indictment parameters $w_n = 1/3$ and $w_n = 1/2$. For the first indictment scheme,

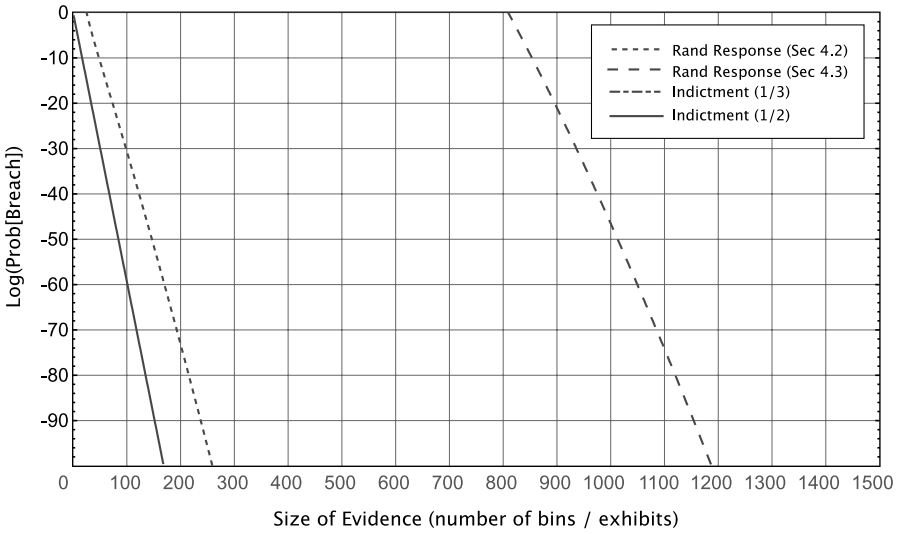

Fig. 2. Data set of 1,000 elements, with required standard deviation 10. The indictment scheme with $w_n = 1/3$ is not visible, as our results do not give non-trivial bounds on the probability of a breach.

we plot the bound given by the Chernoff bound, whereas for $w_n = 1/2$, where all bits must be guessed, we substitute $1/2^n$ for the Chernoff bound.

We consider a simple counting query, in which we want to discover the number of elements in the data set that satisfy a given predicate. In each of the three figures, we will fix the size of the data set, and place a requirement on the accuracy of the final result. This accuracy requirement, articulated by requiring the standard deviation of the result to be at most σ, places constraints on the value of ϵ for ϵ-differential privacy for all schemes as well as the fraction of occupied bins in the randomized response schemes. The current best ϵ-differential privacy approaches for counting, taken from Dwork et al. [2006], add error to the result with standard deviation $1/\epsilon$. Consequently, for the randomized response schemes $\sqrt{p(1-p)s + 1/\epsilon^2}$ must be at most σ, and in the indictment schemes $1/\epsilon$ must be at most σ.

With these constraints in place, we can examine the trade-off between security, measured as the logarithm of the probability that the bond is claimed (assuming no additional leakage by the pollster), and the size requirements in the form of bins k or evidences n. In Figures 2, 3, and 4 we examine these tradeoffs for data sets of sizes 10^3, 10^5, and 10^7, and standard deviation requirements of 10^1, 10^2, and 10^3, respectively.

There are several important observations to make about the relative slopes of the lines. First, the indictment schemes are linear, with slope equal to $(\epsilon - w_n^2/2)$, if the Chernoff bound is used, and $(\epsilon - \ln 2)$ for $w_n = 1/2$. These slopes are independent of the size of the data set. On the other hand, the simple bound for randomized response does shift to the right as we increase the noise and size of the data set. The more advanced bound stays put in the three figures, as the ratio of variance (σ^2) to data set size is maintained. If the data

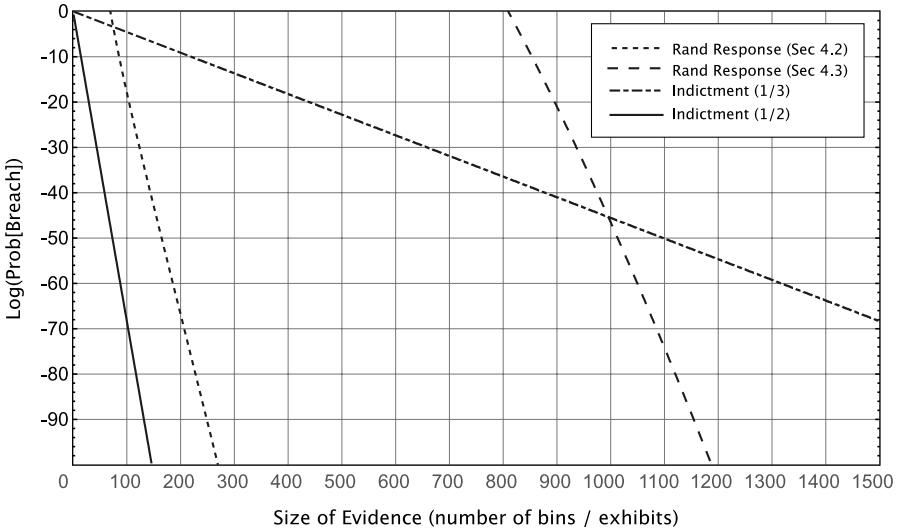

Fig. 3. Data set of 100,000 elements, with required standard deviation 100.

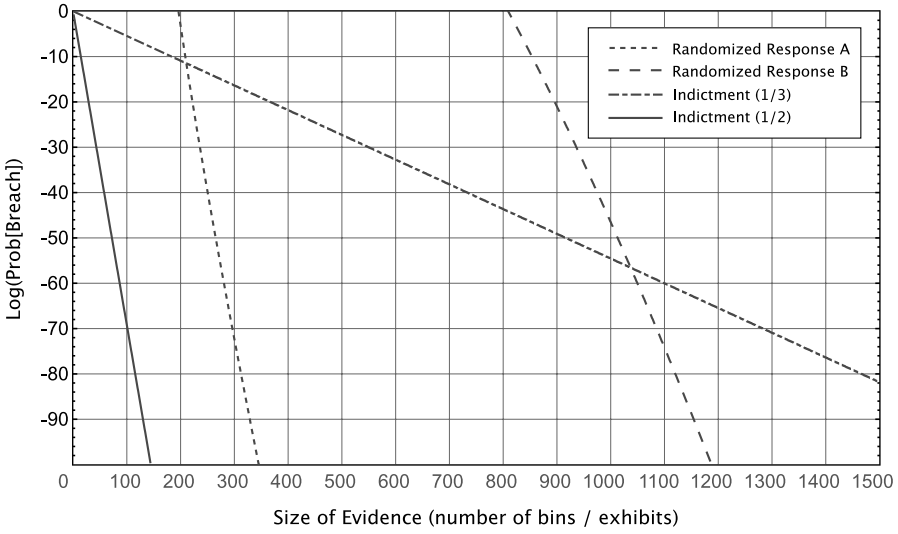

Fig. 4. Data set of 10,00,000 elements, with required standard deviation 1, 000.

set and standard deviation were taken to unreasonable sizes, the advanced bound would surpass the more naive one.

8. CONCLUSION

We have studied four data submission protocols that provide the ability to offer publicly verifiable evidence of data leaks. This evidence is convincing both in that actual leakage can be demonstrated, and in that a fraudulent

indictment is highly unlikely to succeed in the absence of leakage. All four protocols assume the presence of proactive bounty hunters who submit baits to the data collector. Baits are indistinguishable from regular data but offer irrefutable evidence of a data leak when one occurs.

Our four protocols differ in the properties they offer. The first protocol allows for noninteractive bounty verification and relatively exact data collection. The second protocol uses a form of randomized response to collect data, and allows every input to serve as a bait. The third and fourth protocols permit the pollster to publicly disclose a limited amount of non-identifying information about the data collected, but they require an interactive indictment process. The fourth protocol assumes a trusted party charged with reviewing the indictments.

These four protocols demonstrate several desirable properties of a data collection mechanism with self-enforcing privacy. We leave open the problem of designing a protocol that offers all these properties simultaneously. Understanding which features are compatible with others, and which (if any) are mutually exclusive, is an interesting direction for future research.

ACKNOWLEDGMENTS

The authors would like to graciously acknowledge conversations with and comments given by Cynthia Dwork, Moni Naor, and Stephen Fienberg.

REFERENCES

AGRAWAL, R., HAAS, P. J., AND KIERNAN, J. 2003. Watermarking relational data: framework, algorithms and analysis. *VLDB J. 12*, 2, 157–169.

AGRAWAL, R. AND SRIKANT, R. 2000. Privacy-preserving data mining. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'00)*. W. Chen, J. F. Naughton, and P. A. Bernstein Eds. 439–450.

AMBAINIS, A., JAKOBSSON, M., AND LIPMAA, H. 2004. Cryptographic randomized response techniques. In *Proceedings of the Conference on Public Key Cryptography (PKC'04)*. F. Bao, R. H. Deng, and J. Zhou Eds. Lecture Notes in Computer Science, vol. 2947. Springer, 425–438.

BOLDYREVA, A. AND JAKOBSSON, M. 2003. Theft-protected proprietary certificates. In *Proceedings of the Conference on Security and Privacy in Digital Rights Management (DRM'02)*. J. Feigenbaum Ed. Lecture Notes in Computer Science, vol. 2696. Springer, 208–220.

BONEH, D. AND SHAW, J. 1998. Collusion-secure fingerprinting for digital data. *IEEE Trans. Inf. Theory 44*, 5, 1897–1905.

CAMENISCH, J. AND STADLER, M. 1997. Proof systems for general statements about discrete logarithms. Tech. rep. 260, Dept. of Computer Science, ETH Zurich.

CHAUM, D. 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. *Comm. ACM 24*, 2, 84–88.

CHAUM, D. AND PEDERSEN, T. P. 1993. Wallet databases with observers. In *Proceedings of the Conference on Advances in Cryptology (CRYPTO'92)*. E. F. Brickell Ed. Lecture Notes in Computer Science, vol. 740. Springer, 89–105.

CHOR, B., FIAT, A., NAOR, M., AND PINKAS, B. 2000. Tracing traitors. *IEEE Trans. Inf. Theory 46*, 3, 893–910.

DWORK, C. 2006. Differential privacy. Invited talk. In *Automata, Languages and Programming (ICALP2)*. M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener Eds. Lecture Notes in Computer Science, vol. 4052. Springer, 1–12.

DWORK, C., LOTSPIECH, J. B., AND NAOR, M. 1996. Digital signets: Self-enforcing protection of digital information. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC'96)*. 489–498.

DWORK, C., MCSHERRY, F., NISSIM, K., AND SMITH, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC'06)*. S. Halevi and T. Rabin Eds. Lecture Notes in Computer Science, vol. 3876. Springer, 265–284.

GOLDREICH, O., MICALI, S., AND WIGDERSON, A. 1987. How to play any mental game, or a completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC'87)*. 218–229.

GOLLE, P., MCSHERRY, F., AND MIRONOV, I. 2006. Data collection with self-enforcing privacy. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS'06)*. R. Wright, S. D. C. di Vimercati, and V. Shmatikov Eds. 69–78.

GROTH, J. 2002. A verifiable secret shuffle of homomorphic encryptions. In *Public Key Cryptography (PKC'03)*. Y. Desmedt Ed. Lecture Notes in Computer Science, vol. 2567. Springer, 145–160.

JAKOBSSON, M., JUELS, A., AND NGUYEN, P. Q. 2002. Proprietary certificates. In *Topics in Cryptology (CT-RSA'02)*. B. Preneel Ed. Lecture Notes in Computer Science, vol. 2271. Springer, 164–181.

KENTHAPADI, K., MISHRA, N., AND NISSIM, K. 2005. Simulatable auditing. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS'05)*. C. Li Ed. 118–127.

MARGOLIN, N. B., WRIGHT, M., AND LEVINE, B. N. 2004. Analysis of an incentives-based secrets protection system. In *Proceedings of the ACM Workshop on Digital Rights Management (DRM'04)*. A. Kiayias and M. Yung Eds. 22–30.

NEFF, C. A. 2001. A verifiable secret shuffle and its application to e-voting. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS'01)*. 116–125.

OGATA, W., KUROSAWA, K., SAKO, K., AND TAKATANI, K. 1997. Fault tolerant anonymous channel. In *Information and Communication Security (ICICS'97)*. Y. Han, T. Okamoto, and S. Qing Eds. Lecture Notes in Computer Science, vol. 1334. Springer, 440–444.

PFITZMANN, B. AND SCHUNTER, M. 1996. Asymmetric fingerprinting (extended abstract). In *Advances in Cryptology (EUROCRYPT'96)*. U. M. Maurer Ed. Lecture Notes in Computer Science, vol. 1070. Springer, 84–95.

POLLARD, J. M. 1978. Monte Carlo methods for index computation (mod p). *Math. Comput. 32*, 918–924.

SCHNORR, C.-P. 1991. Efficient signature generation by smart cards. *J. Cryptol. 4*, 3, 161–174.

WARNER, S. L. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *Amer. Stat. Assoc. 60*, 309, 63–69.

YAO, A. C.-C. 1982. Protocols for secure computations (extended abstract). In *Proceedings of the IEEE 23rd Annual Symposium on Foundations of Computer Science (FOCS'82)*. 160–164.

Received February 2007; revised July 2007; accepted August 2007