Distributed Privacy-Preserving Mean Estimation

Mirco Schönfeld, Martin Werner
Mobile and Distributed Systems Group
Ludwig-Maximilians-Universität in Munich
mirco.schoenfeld@ifi.lmu.de, martin.werner@ifi.lmu.de

***Abstract*—Due to the rise of mobile computing and smartphones, a lot of information about groups has become accessible. This information shall often be kept secret. Hence distributed algorithms for privacy-preserving distribution estimation are needed. Most research currently focuses on privacy in a database, where a single entity has collected the secret information and privacy is ensured between query results and the database. In fully distributed systems such as sensor networks it is often infeasible to move the data towards a central entity for processing. Instead, distributed algorithms are needed. With this paper we propose a fully distributed, privacy-friendly, consensus-based approach. In our approach all nodes cooperate to generate a sufficiently random obfuscation of their secret values until the estimated and obfuscated values of the individual nodes can be safely published. Then the calculations can be done on this replacement containing only non-secret values but recovering some aspects (mean, standard deviation) of the original distribution.**

I. Introduction

Context-aware computing is a key tool in ubiquitous and mobile computing domains. In ubiquitous computing environments and especially in the envisioned Internet-of-Things (IoT) a multitude of different systems will be able to communicate with each other [1]. However, the scale of these systems will prohibit the central aggregation of data and hence the intelligence has to move towards the data acquisition nodes. It is predicted that the number of sensors will grow quickly [2]. The Internet-of-Things is designed to help users with their respective tasks. As a consequence of context-awareness, computers will have access to a massive amount of private information. Hence, privacy is a prerequisite for context-awareness. This paper adresses the privacy protection in a highly connected distributed system (e.g., the Internet of Things) for the estimation of a probability distribution of some distributed measurements (sensor readings, values, surveys, age, etc.).

In cases, where some query over a distributed dataset needs to be calculated, one has two general choices with respect to the system architecture. Either all private data is collected in a trusted third party performing the needed calculations on the raw data or all data is anonymized before publishing and calculations are performed on this anonymized data [3].

Many existing solutions like [4]–[7] rely heavily on the architecture employing a trusted third party (e.g., a statistical database) collecting information from peers and publishing some statistical information about the complete database to untrusted third parties. In consequence, these algorithms provide differential privacy where the change of a minority of items in the database does not yield a significant change of the outcome of allowed queries. From a privacy perspective, this is a very strong notion of privacy, when the database is actually a central element. However, in ubiquitous computing systems data is often generated in a distributed manner by the nodes of the system. In these situations, a central trusted third party becomes a single point of failure, single point of attack and a bottleneck with respect to data flows. In these situations, a fully distributed privacy-aware architecture is more desirable.

Therefore, we want to provide full privacy and remove the need of a trusted central entity. This state of mind is caused by a common mistrust against any centralized service provider that could arbitrarily gather personalized or sensitive data of any kind. Moreover, for the next decades we expect the number of participants in distributed networks to increase beyound a scale, where a central database computation will quickly become infeasible. Some approaches to assuring ϵ-differential privacy in a distributed setting have been proposed [8], [9]. This approach is also based on adding distributed noise. Our work differs from their work in that we provide a simpler analysis of the impact of randomization on the knowledge gain of an eavesdropper by using Gaussian noise and the standard error of the mean as a measure for information disclosure.

A. Problem Statement

Assume a distributed system that is requested to estimate a distribution of numerical values. Of course, the system could be a distributed database that is queried to aggregate sensitive data. Also, one could assume a service that addresses a questionaire to its users where a question requires an answer in numerical form (e.g., Likert scales). In that case, the service provider is interested in the overall result of the questionaire (i.e., the mean value and its standard deviation) while users want to keep their individual answers private. However, the remaining question is: How can the peers collaborate to provide an estimated distribution without privacy invasion?

II. The Concept

We propose an algorithm that collects answers from all peers in a privacy-preserving manner. The main idea of this approach is a collaborative negotiation of distributions between peers where each peer uses a common error distribution to protect its input to the algorithm from eavesdropping. A distribution of all answers is then calculated from each peers state. As the random influence in this approach is quite high, the system will slowly converge to a consensus and hence, after a sufficient number of rounds of perturbation, no node has

private information and all nodes can take part in a distributed distribution estimation using for example the algorithm of Chan [10]. The proposed approach is composed of three steps:

1) Construction of a privacy-preserving overlay network
2) Several rounds of communication and calculations using this overlay network
3) Distributed or central aggregation of the results

In the first step, the central entity surveying a number of clients helps in the construction of a random network of high node degree between the peers. This network is reconstructed for each round of a fixed number of rounds. In these rounds, some random collection of data is completed. In the final step, the distributed and randomly perturbed data is summarized into the average of all individual inputs. Therefore, the following sections describe the three steps of the algorithm in more detail.

A. Construction of the Overlay Network

Before the overlay network is constructed, all peers taking part in the survey will be connected to a single, central entity, which is reliable for issuing questions as well as for the coordination of the distributed consensus system. This central entity can communicate with all peers. Note that this logically central entity can still be realized in a highly distributed manner. The intended overlay network will be a connected random graph of constant degree m. Consequently, each node can communicate with exactly m neighbors in a private manner in each round.

For private communication, the concept of Locagram Exchangers is used [11]. This concept is based on coupling identity to public keys. Each node generates a public key and registers itself with the platform using this public key. The central service must provide a platform called exchanger, where such Locagrams can be stored and retrieved without any additional authentication. These packages are encrypted for the target node and accompanied by a public key of the target as destination address. In such a system, anyone can see activity (e.g., the fact, that some public key has been addressed by packets). All other information remains confidential. An exchanger can be seen as a micro-blogging platform and could be designed in a way, such that only queries from registered public keys are allowed and that there is a strong limitation on the number or speed of answers to queries using exponential delays, such that one can efficiently make only the platform itself possesses complete activity information.

The overlay graph is now constructed as follows: Each node generates an RSA key pair and publishes his public key on the platform. Each node then downloads all public keys and locally selects a random peer's public key. With this public key and the exchanger, the node can communicate with this peer. We assume, that the communication will be symmetric and hence each peer will send an encrypted pairing message to the selected peer containing the own public key establishing a virtual, symmetric, confidential and authenticated communication channel between two nodes.

This process is performed in a distributed manner such that any node tries to collect m peers. Note that the central exchanger can efficiently limit the number of packets which can be sent to a given peer identified by its public key.

Interestingly, the central entity posesses enough information to calculate connected components of the network and can be allowed to either couple several randomly selected peers from different connected components. However, in real application scenarios the probability of constructing disconnected components is quite low depending on the actual values of the degree m and the number of nodes N and, hence, taking countermeasures against this unlikely situation is unnecessary. In total, the nodes have agreed on a random, fully-connected graph of degree m.

B. Random Perturbation

In general, the algorithm is based on the distribution of very short samples drawn from several specific distributions in a fixed number R of rounds.

The first distribution is given by every nodes current estimation. Every node maintains the current estimated distributions parameters $\mathcal{N}_1 = \mathcal{N}(\mu_E, \sigma_E)$. The estimated mean μ_E is initialized with the correct and secret answer $\mu_E = \mu_S$ and the estimated standard deviation of the current estimation is initialized to $\sigma_E = 0$.

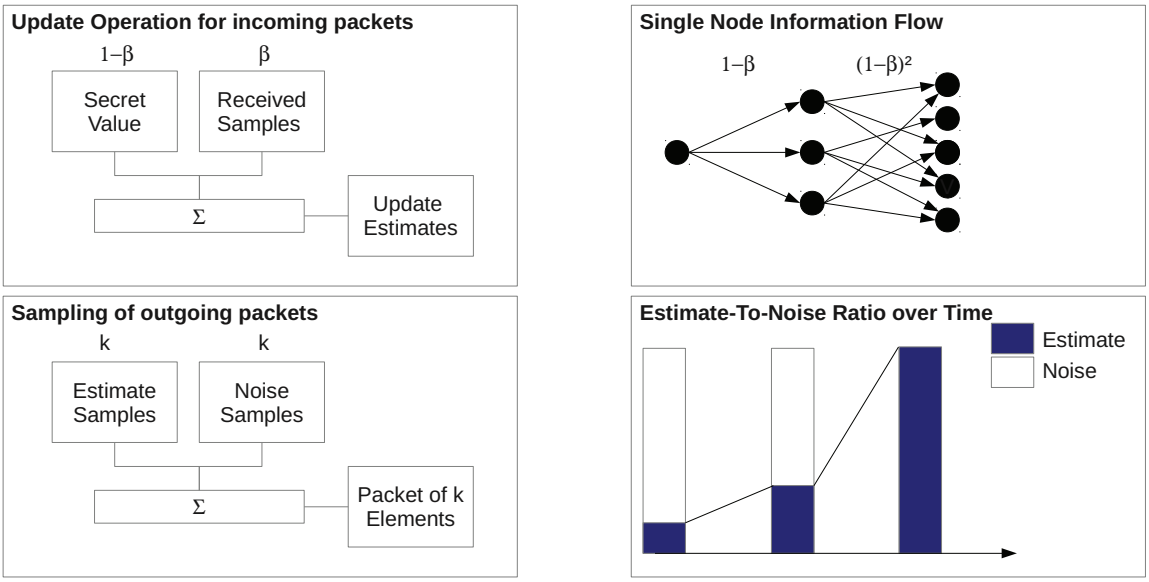
As we intend to construct a privacy-aware algorithm, we have to find a way in which each individual can give input to the algorithm without revealing his own choice. This can be done by adding a common error distribution (random noise) by each individual to his own input and communicating this disturbed input. This motivates the introduction of a second distribution as a zero mean Gaussian distribution with arbitrary high standard deviation $\mathcal{N}_2 = \mathcal{N}(0, \sigma_t)$. This standard deviation is reduced to zero during the first half of the planned rounds. Hence, in the first half of the rounds, a strong perturbation is applied, while in later rounds no additional perturbation is applied anymore. Now, if the variation of the error distribution is large enough and the number of samples per message is sufficiently small, no one can reveal the actual value of an individual from a single message. However, for a multitude of observed messages based on the same value, it is clear, that the effect of the error distribution reduces over time. Therefore, the overlay network can be reconstructed in every round.

These two distributions are used to construct the packets, which are sent by a single peer in each round. A peer constructs a packet by sampling a fixed small number k of samples from sum of the two distributions:

$$\mathcal{N}_1 + \mathcal{N}_2 = \mathcal{N}(\mu_E, \sigma_E) + \mathcal{N}(0, \sigma_t)$$

Note that this sum is preserving the mean. The k values sampled from this distribution are then sent over the network. Of course, in the beginning, when $\mu_E = \mu_S$, the mean of this sample is the secret value μ_S. But chosing k small enough and σ_t large enough results in the expected error of this mean to be arbitrarily high.

For a node receiving a packet, the current estimate of this node is updated as follows: The receiving node takes his own secret value a fixed number of times l. Out of all received samples from all previous transmissions, the last $\frac{\beta}{1-\beta} l$ samples



(a) The update and sampling operation for incoming and outgoing packets performed by each node.

(b) Slow Information Dissemination and Noise Reduction over Time

Fig. 1. The building blocks of the approach

are taken. This results in a sample containing the true value with a fraction of $1 - \beta$ and the received samples with a fraction of β. The updated state is then given by estimating the parameters of this sample. Typically, β will be chosen such that the influence of incoming packets on the estimation is high while the secret information is only marginally put into the distribution. Hence the last received samples will easily contain samples from different peers. Taking into account the current received samples from different peers can be motivated by the result of Agraval that a common, known error distribution can be reconstructed almost perfectly, when enough samples of this distribution are collected [12]. As a consequence, an attacker eavesdropping many packets will eavedrop uncertain knowledge about a non-constant value containing only partial secret information depending on the value of β. Figure 1(a) depicts the two isolated parts of the algorithm: The update of the estimated distribution from incoming packets and the generation of outgoing packets. Figure 1(b) shows the slow distribution of information as controlled by β and the estimation to noise ratio over time. Note that finally, when no noise is applied, the value of the estimate will randomly differ from the secret value.

In summary, the overall knowledge over all peers converges to the true answer while, in contrast to distributed consensus algorithms like [13], a single peer is prevented from calculating the correct answer. In certain scenarios like conducting surveys this is a desirable side-effect, especially, if complete results of such a survey should not be revealed to an attacker.

C. Aggregation

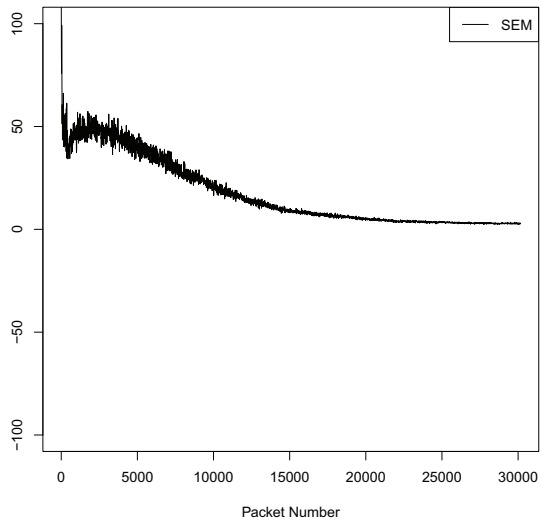
As the main step of the algorithm is a sufficiently random procedure assigning completely different values to each node in a mean-preserving way, the time to convergence is rather high. Hence, we propose to stop the process after a fixed time and publish all estimated means μ_E of the individuals, as long as they changed enough. The mean of these individual estimations is then the result of the algorithm. As the random influence on the current state of all nodes is high, one does not need a trusted central service and can readily apply a distributed mean estimation algorithm.

Altogether, we have shown a way to collect the desired information without enabling a central entity to violate any user's privacy. In fact, a user's privacy is also protected against other peers of the network as the knowledge of a single peer does not suffice to disclose the information of any other peer.

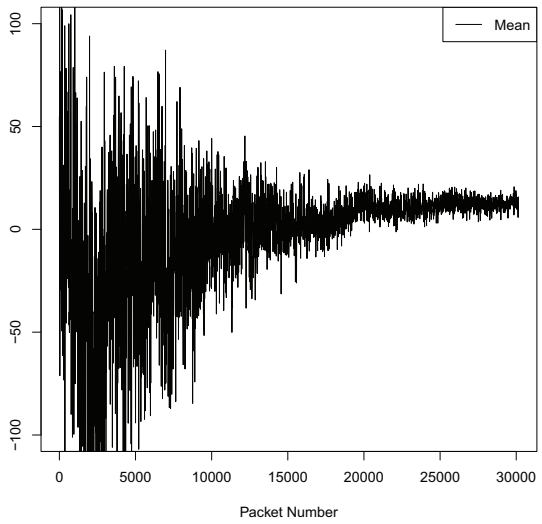
III. Simulations

We conducted several simulations with changing mean and standard deviation of the target distribution and randomly assigned true values to the individual peers. The first observation is that the final estimate of the algorithm is fairly good taking into account how strong the individual packets have been randomly obfuscated. Figure 3 depicts the quality of a series of simulations with varying means and standard deviations. As can be clearly seen, the final outcome of the algorithm quite accurately approaches the optimum $f(x) = y$.

Figure 2 shows the results of the algorithm over time. Figure 2(a) shows the standard error of the mean, which is a measure for the ability to derive a mean from an observation. A detailed discussion of this measure with respect to privacy is given in the following section. Figure 2(b) shows the overall estimated mean over time, which is the result of applying the aggregation step at each round. What can be clearly seen from this figure is how the variance of the mean is reducing over time as a result of reducing the influence of the distortion



(a) SEM of observed packets over time



(b) Estimated Sample Mean of Observed packets over time

Fig. 2. Standard error of the mean (SEM) scaling and SEM over time

distribution $\mathcal{N}_2$ over the rounds. It is worth noting that an optimal attacker would estimate these values from observing the communication.

IV. Privacy of Distributed Consensus Questionaire

The main concern of this paper is the privacy of the individual with respect to

- its peers,
- the crowd,
- and a central exchanger.

Therefore, we will enumerate the complete information flow inside this network. We differentiate between meta-information and service information. Meta-information shall be any publicly observable information about network activity including addressed peers, technical addresses, timestamps. Service information is limited to the actual algorithmic inputs and is the main goal of protection.

Meta-information is generated by the system in different phases. In the first phase, a complete view of the network node set is constructed by the central collection of public keys. As the public keys are regenerated for each questionaire, they do not contain information but are rather random numbers. The central entity, however, can collect bindings of technical addresses and public keys. As no service information is exchanged unencryptedly, the impact of this meta-information on user privacy is limited to attacks with external sources of knowledge, which can never be prevented without a truly confidential communication system.

Information is generated and communicated only between up to m random peers. Moreover, this information is perturbed with pairwise uncorrelated errors drawn from a commonly used error distribution. As shown by Agrawal and Aggarval, this approach gives good privacy gain as long as the number of observable packets is low [12]. Furthermore, only in the first round, each node communicates its own secret information with strong perturbation. In later rounds of the distributed algorithm, a sample is communicated based on the values of the neighbors and a merely small portion of confidential information. Altogether, the systems achievement can be subsumed as follows: The work of this privacy-friendly cooperative mean estimation is fully distributed and hence the system scales arbitrarily as long as the final mean estimation is also performed in a fully distributed way.

A. Privacy Measurements

For a detailed discussion of the system's privacy, we first note, that the mean of early observed packets will be a very good estimator of the value of an individual. As each individual starts off with a distribution with his individual value as the mean and a zero standard deviation and as the distortion has zero mean the sum of these two distributions has the secret individual value as the mean and the sum of the variances as the variance, as the two distributions are independent from each other.

The quality of deriving the mean out of a finite sample is given by the standard error of the mean (SEM). The standard error of the mean is defined to be the standard deviation of the differences of the true mean and all means that will occur when a fixed number of n samples are drawn from a distribution. The standard error of the mean is hence decreasing with the size of the sample and increasing with the standard deviation of the distribution. Fortunately, our protocol arranges the situation that the number of observable samples is small

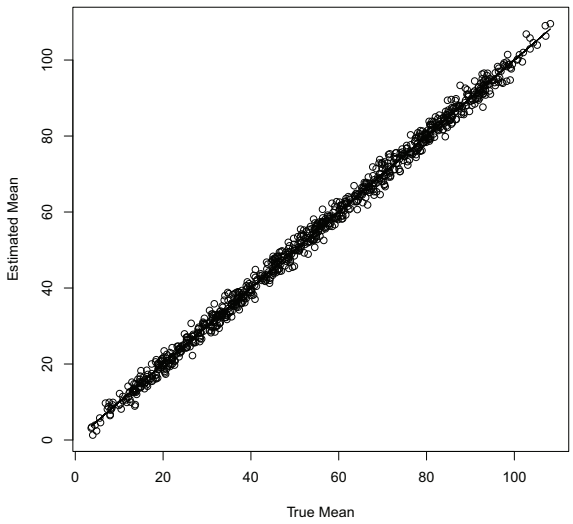


Fig. 3. Estimated Means vs. True Means

and the standard deviation of the combined distribution is large.

The privacy of this approach can be explained by the following well-known fact: Decreasing the standard error of the mean by a factor of ten needs the number of samples to be increased by a factor of hundred.

Note that the proposed protocol does not generate unlimited samples of the objective distribution (the initial distribution of a node), as the incoming information is used to drift the distribution towards the global mean.

From the standard error of the mean one can derive confidence intervals and can define probabilistic attacks, which are already successful, when the true value is limited to a small-enough interval with acceptable probability.

Define a probabilistically successful attack to be an attack in which with a given fixed error probability p the mean can be bound to an interval of length at most l. This length can then be given in terms of the standard error of the mean. Taking the $(1-p)$-percentile point z of the standard normal distribution, we can conclude that with a probability of $1-p$ the true value of the observed mean lies inside the interval

$$[\bar{x} - (z * \text{SEM}), \bar{x} + (z * \text{SEM})]$$

Figure 4(a) shows a plot of the factors z on the Y-axis and the probability of error on the X-axis. For a high confidence of 95% the value is $z \approx 1.96$, hence the length can be bound to an interval of length $3.92 * \text{SEM}$. For a low confidence of 50% the value is given by $z \approx 0.675$ and the interval is bound to the length $1.35 * \text{SEM}$. In total, a sufficiently large standard error of the mean provides arbitrary privacy.

V. Privacy Risks and Attacks

This section gives a detailed view on several possible attacks against the proposed scheme. It will be clarified what an attacker would need to know or to do to gain knowledge of single peer's private information, to gain knowledge of the overall result of a survey, or to influence the result of a survey in a certain way.

The scenario in which an attacker controls every peer is considered trivial, since then he is able to control the survey in any way or gain any knowledge of every peer.

A. Eavesdropping

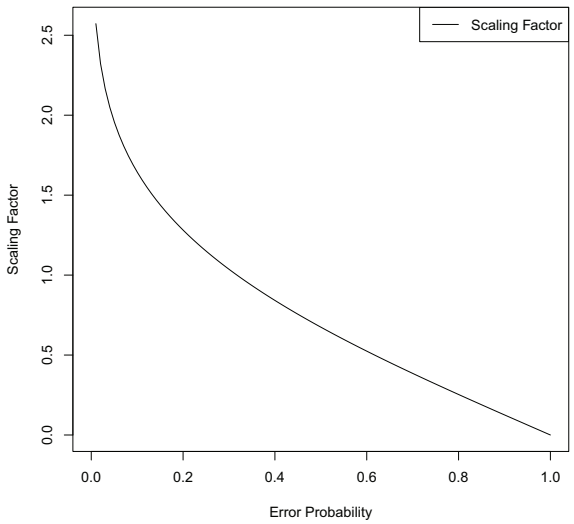
One prevalent form of attacking a distributed system is eavesdropping messages between peers. In this scenario, an adversary would try to unveil the secret value of one single peer or the overall result of a survey.

The distribution an attacker would observe at one point in time has a mean given by the following formula

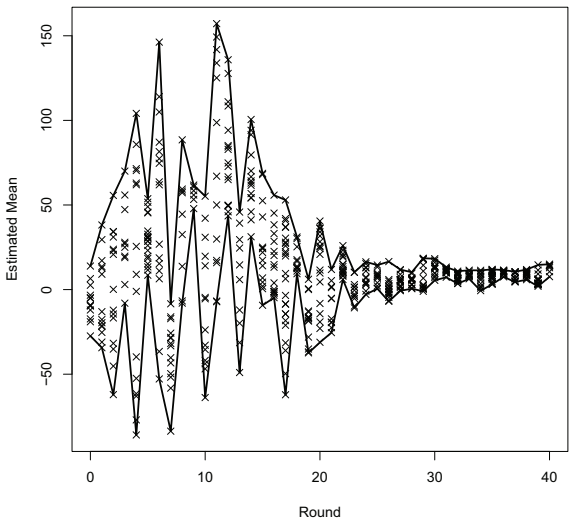
$$\mu_c = \beta\mu_R + (1-\beta)\mu_s$$

where μ_R stands for the mean value that one peer calculates from a received distribution and μ_s for its own secret value. Meanwhile, β controls the ratio between secret and received samples a peer would use to re-create a distribution with mean μ_c that he then uses to communicate a sample of it to its neighbors. Hence, revealing a peer's secret value means to estimate μ_s from the samples of the observable distribution with mean μ_c.

Estimating μ_s is nearly impossible as the standard error of the mean of communicated packets is high. Also, the more packets an attacker eavesdrops, the higher the observable variance of the packets, since the mean values μ_R and μ_s will increasingly differ. The only information an attacker



(a) Scaling factors for confidence intervals of the SEM



(b) Estimated means μ_E of a single node during each round of the algorithm. For each round, all intermediate estimates from incoming packets are depicted. While the secret value of this node is -11 the packets drifted quickly and significantly towards a completely different value.

Fig. 4. Scaling Factors for the standard error of the mean and the result of eavedropping a single nodes estimates

could estimate by continuous observation is the overall result of a survey. But, the result that could be observed from eavesdropping only one peer's communication will not match the actual result, since that has to be calculated from every peer's contribution. Figure 4(b) depicts the estimates μ_E over time of a single sensor node in a simulation with 50 nodes, 40 rounds, $\beta = 0.9$, and a secret value of the node $\mu_S = -11$. While the complete system estimates a final mean value of 10.35, the mean of the single node is quite variable over time. The scatter points show estimates including the effect of incoming packets in each round and hence, as there is no time-synchronization, these are the possible estimated distributions from which outgoing packets might be generated.

However, if an attacker is able to eavesdrop one peer's packets from the first round, a good estimator for the secret value is given by the sample mean. However, the standard error of the mean of a sample of length k is given by

$$SEM = \frac{\sigma}{\sqrt{k}}$$

and for the case of our algorithm $\sigma \approx \sigma_t$ is large for the early rounds.

Altogether, an eavesdropper can either observe the correct mean with a high standard error of the mean in early rounds or another value randomly influenced from peer data, which only contains the secret value in a marginal fraction. Hence, the secret value is never unveiled to eavesdroppers with sufficient confidence.

B. Isolation Attack

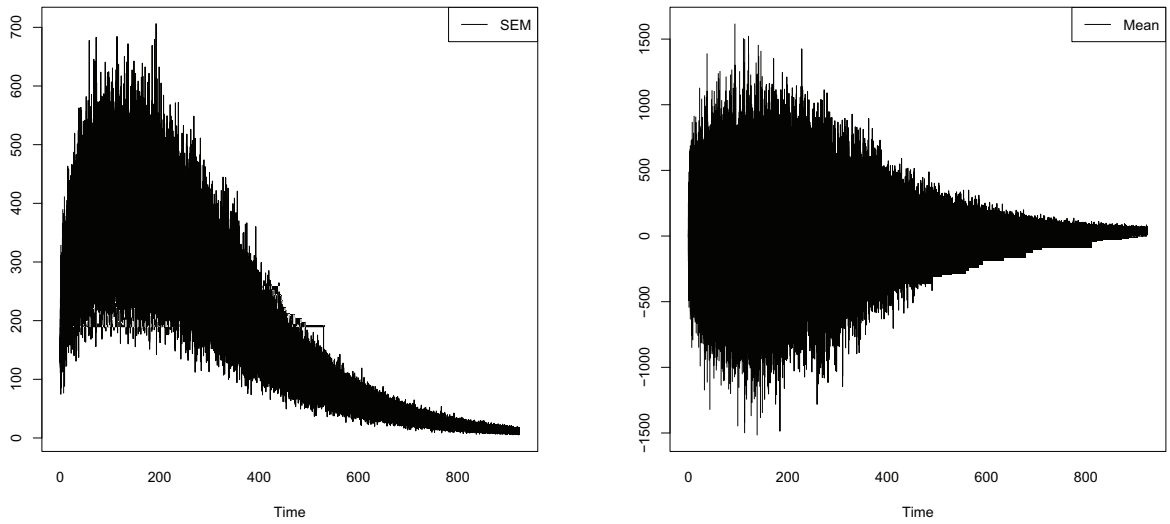
An isolation attack aims solely on unveiling the answer of one single peer. While this technique promises very accurate results, it is merely impossible to achieve.

From the definition of μ_c given above, it is easy to deduce that the secret value μ_s can be calculated if μ_R is known by an attacker. Meanwhile, isolating a peer means to replace all of its communication partners controlling all of its communication and thereby providing him with several known μ_R. In general, an attacker creates a distribution with a known mean and sends it to its victim. The victim responds with a distribution that integrates its own answer and a perturbation assumed to be sufficient. But, the attacker is able to extract the perturbation, since it is exactly the μ_R he himself provided. The success of the attack depends on the number of packets an attacker is able to inject as well as on the number of observed packets resulting from this influenced distribution.

While this scenario sounds dangerous to any peer's privacy, it is ruled out by design. Unless the attacker is able to recreate the whole survey system architecture, the proposed overlay network and corresponding enforcement of communication pairing efficiently inhibits isolation attacks. Especially, the choice of peers is taken locally from a large list of public keys. As long as the atacker does not control this list and hence a majority of the network, the probability is very high, that the nodes obtain information from non-malicious peers.

C. Influence of a Minority

Influence of a minority on the result of a survey is a critical question. However, the problem lies outside the scope of the



(a) SEM of observed packets over time

(b) Estimated Sample Mean of Observed packets over time

Fig. 5. Real-World Survey results for age estimation of a group of radio listeners

actual survey system as long as privacy shall be maintained. Privacy basically implies anonymity and opens doors for an atacker to control a multitude of malicious nodes taking part in the survey. Hence, the survey can be influenced by its design. It is an interesting research direction to integrate countermeasures based on techniques such as proof of work, anomaly detection, captchas or similar anonymous proofs of being non-malicious.

VI. Application

A real-world experiment has been conducted with actual survey data about the age of radio listeners in Bavaria. The survey consisted of 1761 participants with ages ranging from 10 to 69 years performed in an one-hour interval between 6 a.m. and 7 a.m.. This survey has been used to initialize the nodes. The obfuscating distribution was set to $\mathcal{N}_2 = \mathcal{N}(0, 600)$, the system performed 20 rounds, a packet contained 15 samples, the informaton control parameter β was set to 0.9 and every node chose 25 peers per round as destinations. From this setting, we can conclude for the standard error of the mean of the first packet

$$\text{SEM} \geq \frac{600}{\sqrt{15}} \approx 154.9$$

As can be seen from Figure 5, the standard error of observed packets takes similar behaviour compared to the simulation results and is rather high compared to the actual data range of an age survey. This standard error of the mean effectively renders a possible attacker to be no better than randomly chosing an average age of 50 years. On the other hand, after enough perturbation has taken place, the mean quickly stabilizes. Our proposed algorithm estimates a mean of 38.886 for a true value of 37.522.

VII. Conclusion

With this paper, we have constructed a system for privacy-friendly estimation of a gaussian distribution with the following key features: Its privacy is *measurable* and *configurable* based on the standard error of the mean, it can be used to *anonymize* the individual secret values in a fully distributed way with low computational overhead for the individual nodes and this anonymization is itself *distributed* The main drawback of the system at hand is, that the basic communication system should be a highly small-world communication graph where sufficiently many random peers can be selected as neighbors. This is concretely realized as a peer-to-peer overlay network. However, an interesting research direction would be the analysis of privacy in a more *local* deployment, where the nodes can only communicate with a fixed small subset of the complete communication graph (e.g., nodes that are only few hops away in a wireless mesh network). This leads to more complexity in situations, where the secret values are not distributed randomly over the network neighbours, but have several clusters coming from social groups (e.g., people of different age form different geographic clusters). Further research should try to integrate completely distributed algorithms such as this proposed scheme with more generic systems such as statistical databases. Overall, a distributed database covering a specific set of queries in a privacy-preserving manner would be a perfect match removing the need for the often-applied trusted third party in application domains, where the massive amount of data or the energy consumption of communication prohibits centralization of data.

References

[1] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," 2013, pp. 1–41 (preprint).

[2] A. Zaslavsky, C. Perera, and D. Georgakopoulos, "Sensing as a service and big data," *International Conference on Advances in Cloud Computing (ACC-2012)*, 2013.

[3] S. Goryczka, L. Xiong, and B. C. Fung, "m-privacy for collaborative data publishing," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference on*. IEEE, 2011, pp. 1–10.

[4] L. Fan and L. Xiong, "An adaptive approach to real-time aggregate monitoring with differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2013.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*. Springer, 2006, pp. 265–284.

[6] X. Xiao, G. Bender, M. Hay, and J. Gehrke, "ireduct: differential privacy with reduced relative errors," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM, 2011, pp. 229–240.

[7] Y. Xiao, L. Xiong, and C. Yuan, "Differentially private data release through multidimensional partitioning," in *Secure Data Management*. Springer, 2010, pp. 150–168.

[8] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology-EUROCRYPT 2006*. Springer, 2006, pp. 486–503.

[9] A. Beimel, K. Nissim, and E. Omri, "Distributed private data analysis: Simultaneously solving how and what," in *Advances in Cryptology–CRYPTO 2008*. Springer, 2008, pp. 451–468.

[10] T. F. Chan, G. H. Golub, and R. J. LeVeque, "Algorithms for computing the sample variance: Analysis and recommendations," *The American Statistician*, vol. 37, no. 3, pp. 242–247, 1983.

[11] M. Werner, "Privacy-protected communication for location-based services," *Security and Communication Networks*, 2011.

[12] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, New York, NY, USA, Jan. 2001, pp. 247–255.

[13] A. Speranzon, C. Fischione, and K. H. Johansson, "Distributed and collaborative estimation over wireless sensor networks," in *Decision and Control, 2006 45th IEEE Conference on*, San Diego, CA, USA, Jan. 2006, pp. 1025–1030.