ODO Protocol for Privacy-Preserving [1]

Seminar: Cryptographic Protocol
Karolina Kopyczynska

1 Introduction

Many papers in the cryptography and database communities deal with the problem of statistical disclosure control by preserving the privacy of individuals. It is a currently pressing issue. It extends for example, to the question of how to keep information about hospital patients secret in spite of appraisal statistics about the medical condition.

The individual values are stored in so-called statistical databases that are used for statistical analysis purposes. To guarantee preservation privacy it is common to forbid query access to individual records. To achieve this, only the use of statistical summary information (COUNT, SUM, AVERAGE, MAX, MIN) is allowed and all users queries are audited to ensure that the system is used correctly.

However this is not enough to secure the individual information, since with intelligent use of combination of aggregate queries it could be possible to discover information about a single individual. One possible solution to keep the private information secret could be to add some noise to the answer of the query. If the answer contains no information regarding individual privacy, this noise will be too 'small' to change the statistic value. But if the answer contains the private information (for example in case, the answer contain the information only about one row) the noise will be sufficient to change it. The purpose of noise generation is to create a distributed implementation of privacy-preserving statistical databases.

In these databases, privacy is obtained by perturbing the true answer to a database query by the addition of a small amount of Gaussian distributed random noise. The generation of Gaussian noise introduces a technique for distributing shares of many unbiased coins.

2 Structure of ODO (Our Data, Ourselves) Protocol

Let database be a collection of rows as an n-tuple $(d_1, d_2, ..., d_n)$ of elements from D. D could be points in $\mathbb{R}^k$, text strings, images, or any other imaginable set of objects. The elements d_i are independent, meaning that revealing one to the adversary would not give information about another. And let the query be a function f mapping rows to the interval $[0, 1]$. The *true answer* to the query is the value obtained by applying f to each row and adding the results.

[1] This work is an enhancement of Dwork 'ODO Our Data Ourselves: Privacy via Distributed Noise Generation'. It shows only Gaussian Noise and it hopefully more understandable for ordinary person.

To emulate any privacy mechanism many powerful techniques of secure function evaluation exist, but generic computations can be expensive. Perturbation of the true answer by adding noise to a database query is inspired by the combination of the simplicity of securely computing sums and the power of the noisy sums. It is no longer necessary to have a central trusted server. The parties can hold their own data in order to act autonomously, while simultaneously preventing malicious parties from interfering with the utility of the data. The ODO protocol assumes that every data holder participates in every query and that the function f is predicate, meaning that the approach of decentralization is quite simple.

The ODO protocol according to Dwork:

Structure of ODO (Our Data, Ourselves) Protocol

1. **Share Summands:** On query f, the holder of d_i, the data in row i of the database, computes $f(d_i)$ and shares out this value using a *non-malleable verifiable secret sharing scheme*. The bits are represented as 0, 1 values in Galois Field GF(q), for a large prime q. We denote this set $\{0,1\}_{GF(q)}$ to make the choice of field clear.

2. **Verify Values:** Cooperatively verify that the shared values are legitimate (that is, in $\{0,1\}_{GF(q)}$, when f is a predicate).

3. **Generate Noise Shares:** Cooperatively generate shares of appropriately distributed random noise.

4. **Sum All Shares:** Each participant adds together all the shares it holds, obtaining a share of the noisy sum $f(d_i)$+noise. All arithmetic is in GF(q).

5. **Reconstruct:** Cooperatively reconstruct the noisy sum using the reconstruction technique of the verifiable secret sharing scheme.

3 Tools

3.1 Terminology

For proper understanding, it is necessary to unify common terminology.

Values *in shares* are shared and verified, but not reconstructed. Values that are publicly known are said to be *public*.

An Extractor is a function, that is used to extract randomness from weakly random sources. An *randomness extractor* [1] is a method of converting a non-uniform input distribution into a near-uniform distribution on a smaller set.

The *Shannon information* content of an outcome $\boldsymbol{x}$ is defined to be

$$h(x) = log_2 \frac{1}{P(x)}$$

It is measured in bits.

The *entropy* (in information theory) is a measure of the uncertainty associated with a random variable. The entropy of an ensemble X is defined to be the average Shannon information content of an outcome:

$$H(X) \equiv \sum_{x \in A_X} P(x) log \frac{1}{P(X)}$$

When it is convenient, we may also write $H(x)$ as $H(p)$, where p is the vector $(p_1, p_2, ...p_n)$. Entropy is maximized if p is uniform [2].

Definition *Letting the min-entropy of a distribution D on X be denoted* $H_\infty(D) = -logmax_{x \in X} D(x)$. *A function* $F : X \times Y \rightarrow \{0,1\}^n$ *is a* (δ, ϵ, n)*-extractor, if for any distribution D on X such that* $H_\infty(D) > \delta$,

$$|\{F(x,y) : x \in_D X, y \in_U Y\} - U_n| < \epsilon$$

where $|.|$ *is the statistical distance between two distributions,* U_n *is the uniform distribution on* $\{0,1\}^n$, *and* $x \in_D X$ *stands for choosing* $x \in X$ *according to D.*

This means that the input distribution has sufficiently high min-entropy, a good extractor takes a short seed and outputs a distribution that is statistically close to the uniform.

A *privacy mechanism* is an interface between a user and data. It can be interactive or non-interactive.

A mechanism gives ϵ-*indistinguishability* [3] if for any two data sets that differ on only one row, the respective output random variables (query responses) τ and τ^* satisfy the following for all sets S of responses:

$$Pr[\tau \in S] \leq exp(\epsilon) \times Pr[\tau^* \in S].$$

Similarly, a mechanism gives δ-*approximate* ϵ-*indistinguishability* if for outputs τ and τ^* based, respectively, on data sets differing in at most one row,

$$Pr[\tau \in S] \leq exp(\epsilon) \times Pr[\tau^* \in S] + \delta.$$

The presence of a non-zero δ permits us to relax the strict relative shift in the case of events that are not especially likely.

3.2 Binomial and Gaussian distribution

3.2.1 Binomial distribution

The most common situation types modeled in the theory of probability are *Bernoulli trials*. A Bernoulli trial is an experiment that can be either a 'failure' or a 'success' and the outcome is random. The *Bernoulli random variable*, is just a sum of the number of successes in a single trial.

Binomial distribution plays the central role in probability theory, as a model of one of the most common situations, namely of a sum of the total number of successes in n Bernoulli trials. Let S_n denote the binomial random variable, and use the representation

$$S_n = X_1 + ... + X_n$$

where $X_1, ..., X_n$ are the Bernoulli random variables describing the outcomes of successive trials (i.e., $X_i = 1$ or 0, depending whether the ith trial results in success or in failure). The probability of an single success is given by p. The probability of getting exactly k successes in n trials is given by the probability mass function:

$$P\{S_n = k\} = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k} = b(k; n, p)$$

mean

$$E(S_n) = np,$$

and the variance

$$Var(S_n) = np(1-p).$$

3.2.2 Gaussian (Normal) distribution

The Gaussian distribution is one of the most, elsewise main distributions in both probability theory and statistics, as well as in nature. The density of Gaussian Distribution is defined by

$$f(x) = f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

where μ and $\sigma > 0$ are two parameters, where μ is an arithmetical mean and σ is a variance.

Definition The term ***central limit theorem*** (CLT) is a generic name used to designate any of the series of theorems which assert that the sums of large numbers of random variables after standardization (i.e., subtraction of the mean and division by standard deviation) have approximately the standard normal distribution.

For formal proof see f.e. in [5]. According to CLT it can be proved that binomial distribution can be an approximation of Normal (Gaussian) distribution.

3.2.3 Math for Gaussians and Binomials noise

In [3] it was shown, how to correctly choose the values of ϵ and δ for the Gaussian and Binomial distribution such that the noisy sums primitive yields δ-approximate ϵ-indistingishability. The following determines the largest value of x such that a relative bound of $\exp(\epsilon)$ holds and then integrates the probability mass outside of this interval.

Let τ be an output summed with noise $\tau = \sum_i f(i, d_i)$+noise. Replacing database D with $D^{'}$, where $D^{'}$ differs from D only in one row, so that the summation changes by at most 1. According to definition of ϵ-indistinguishability, the probability that τ occurs with inputs D with noise= x is divided into the probability that τ occurs with inputs $D^{'}$ with noise = $x+1$ should be smaller that $exp(\epsilon)$.

The Gaussian density function is defined as $exp(-x^2/2R)$ so that the probability is define as $Pr(x) \propto exp(-x^2/2R)$. The ratio of densities at two adjacent integral points is

$$\frac{exp(-x^2/2R)}{exp(-(x+1)^2/2R)} = exp(x/R + 1/2R).$$

This value is smaller than $exp(\epsilon)$ for x smaller than $\epsilon R - 1/2$. Consequently, if $R \geq 2log(2/\delta)/\epsilon^2$ and $\epsilon \geq 1$, the integrated probability beyond this point will be at most

$$Pr[x > \epsilon R - 1/2] \leq \frac{exp(-(\epsilon R)^2/2R)}{(\epsilon R)\sqrt{\pi}} \leq \delta.$$

Followed δ-approximate ϵ-indistinguishability will be obtained when R is at least $2log(2/\delta)/\epsilon^2$.

For the Binomial noise let be the probability $p = 1/2$ and the density at point $n/2 + x$. So that

$$Pr[n/2 + x] = \left(\begin{array}{c} n \\ n/2 + x \end{array} \right) 1/2^n$$

As a consequence, relative probabilities are

$$\frac{Pr[n/2+x]}{Pr[x/2+x+1]} = \frac{\left(\begin{array}{c} n \\ n/2 + x \end{array} \right)^{1/2^n}}{\left(\begin{array}{c} n \\ n/2 + x + 1 \end{array} \right)^{1/2^n}} = \frac{n/2+x+1}{n/2-x}.$$

So long as x is no more than $\epsilon n/8$, this should be no more than $(1 + \epsilon) < exp(\epsilon)$.

Due to Chernoff bound, that says

Theorem 1.1 Let $X_1, ..., X_n$ be discrete, independent random variables such that $E[X_i] = 0$ and $|X_i| \leq 1$ for all i. Let $X = \sum_{i=1}^{n} X_i$ and let σ^2 be the variance of X. Then

$$Pr[|X| \geq \lambda\sigma] \leq 2e^{-\lambda^2/4}$$

for any $0 \leq \lambda \leq 2\sigma$

For the Binomial distribution with probability $p = 1/2$ Chernoff bounds can be define as

$$P = \sum_{i=\lfloor n/2 \rfloor+1}^{n} \left(\begin{array}{c} n \\ i \end{array} \right) p^i (1-p)^{n-i} \geq 1 - e^{-2n(p-1/2)^2}$$

Followed for $x < \epsilon n/8$ the probability that a sample exceeds is

$$\begin{aligned} Pr[y > n/2 + \epsilon n/8] &= Pr[y > (1+\epsilon/4)n/2] \\ &\leq exp(-(\epsilon^2 n/64)). \end{aligned}$$

We get δ-approximate ϵ-indistinguishability so long as n is chosen to be at least $64log(2/\delta)/\epsilon^2$. This exceeds the estimate of the Gaussian due to approximation error, and general slope in the analysis, though it is clear that the form of the bound is the same.

3.3 Model of computation

We assume the standard synchronous model of computation in which n processors communicate by sending messages via point-to-point channels and up to $f \leq \lfloor \frac{n-1}{3} \rfloor$ may fail in an arbitrary, Byzantine, adaptive fashion. If the channels are secure, than the adversary may be computationally unbounded. However, if the secure channels are obtained by encryption then it can be assume the adversary is restricted to probabilistic polynomial time computations [1].

Definition The Verifiable Secret Sharing (VSS)

A VSS scheme allows any processor distribute shares of a secret, which can be verified for consistency. If the shares verify, the honest processors can always reconstruct the secret regardless of the adversary's behavior. Moreover, the faulty processors by themselves cannot learn any information about the secret. A nonmalleable VSS scheme ensures that the values shared by a nonfaulty processor are completely independent of the values shared by the other processors; even exact copying is prevented.[1]

3.4 Adaptive Query Sequences

What happened after multiple queries? Degrade the values of ϵ and δ in unintentional manner?

Theorem 1.2 *A mechanism that permits T adaptive interaction with a δ-approximate ϵ-indistinguishable mechanism ensures δT-approxiamte ϵT-indistinguishability.*

Proof. Started by examining the probability that the transcript, written as an ordered T-tupel, lands in a set $\boldsymbol{S}$.

$$Pr[x \in S] = \prod_{i \leq T} Pr[x_i \in S | x_1, ..., x_{i-1}]$$

As the noise is independent at each step, the condition on $x_1, x_2, ..., x_{i-1}$ only affects the predicate that is asked. As a consequence, can substitute

$$\prod_{i \leq T} Pr[x_i \in S_i | x_1, ..., x_{i-1}] \leq \prod_{i \leq T} (exp(\epsilon) \times Pr[x_i^{'} \in S_i | x_1, ..., x_{i-1}] + \delta)$$

If look at the additive contribution of each of the δ terms, of which there are T, it can be notice that there are only ever multiplied by probabilities, which are at most one. Therefore, each contributes at most an additive δ.

$$\begin{aligned} \prod_{i \leq T} Pr[x_i \in S_i | x_1, ..., x_{i-1}] &\leq \prod_{i \leq T} (exp(\epsilon) \times Pr[x_i^{'} \in S_i | x_1, ..., x_{i-1}]) + \delta T \\ &= (exp(\epsilon T) \times \prod_{i \leq T} (Pr[x_i^{'} \in S_i | x_1, ..., x_{i-1}]) + \delta T \\ &= (exp(\epsilon T) \times Pr[x_i^{'} \in S] + \delta T \end{aligned}$$

The proof is complete.

4 Generating Gaussian Noise [2]

The first strong positive results for output perturbation added noise drawn from a Gaussian distribution, with density function $Pr[x] \propto exp(-x^2/2R)$. In order to recast those results in terms of indistinguishability was shown in Section 3.2 that the addition of Gaussian noise gives δ-approximate ϵ-indistinguishability for the noisy sums primitive when $\epsilon > [log(1/\delta)/R]^{1/2}$. In a similar vein, Binomial noise, where n tosses of an unbiased 1 coin are tallied and divided by 2, also gives δ-approximate ϵ-indistinguishability so long as the number of tosses n is at least $64log(2/\delta)/\epsilon^2$.

The follow solution use the full power of coin-flipping and is cost effective when c is sufficiently large ($\in \Omega(n)$). As a result, it will be required only $\Omega(c)$ sharing of values in GF(2) when $c \in \Omega(n)$. Let n denote both the number of players and the desired number of coins. And according to Dwork [1]:

1. Each player i shares a random bit by sharing out a value $b_i \in \{0,1\}_{GF(q)}$, using a non-malleable verifiable secret sharing scheme, where q is sufficiently large, and engages in a simple protocol to prove that the shared value is indeed in the specified set. (The verification is accomplished by distributively checking that $x^2 = x$ for each value x that was shared, in parallel. This is a single secure function evaluation of a product, addition of two shares, and a reconstruction, for each of the n bits b_i.) This gives a sequence of low-quality bits in shares, as some of the shared values may have been chosen adversarially. (Of course, the faulty processors know the values of the bits they themselves have produced.)

[2] This complete chapter ist cited from [1]

2. Now, suppose for a moment that we have a public source of unbiased bits, $c_1, c_2, ..., c_n$. By XORing together the corresponding $b's$ and $c's$, we can transform the low quality bits b_i (in shares) into high-quality bits $b_i \otimes c_i$, in shares. (Again, the faulty processors know the values of the (now randomized) bits they themselves have produced.) The XORing is simple: if $c_i = 0$ then the shares of b_i remain unchanged. If $c_i = 1$ then each share of b_i is replaced by one minus the original share.

3. Replace each share s by *2s-1*, all arithmetic in GF(q). This maps shares of 0 to shares of -1, and shares of 1 to (different) shares of 1.

4. Finally, each participant sums her shares to get a share of the Binomial noise.

Now it is left to explain how to generate the c_i. Each participant randomly chooses and non-malleably verifiably shares out two bits, for a total of 2n low-quality bits ($b_1^{'}$,$b_2^{'}$,..., $b_{2n}^{'}$) in shares. The $b_i^{'}$ are then reconstructed, so that they become public. The sequence $b_1^{'}b_2^{'}...b_{2n}^{'}$ is a bitfixing source: some of the bits are biased, but they are independent of the other bits (generated by the good participants) due to the non-malleability of the secret sharing. The main advantage of such a source is that it is possible to apply a deterministic extractor on those bits and have the output be very close to uniform. Since the bits $b_1^{'}b_2^{'}...b_{2n}^{'}$ are public, this extraction operation can be done by each party individually with no additional communication. The currently one of best known deterministic extractor is describe in [4], which produces a number $m > n$ of nearly unbiased bits. The outputs of the extractor are public coins $c_1...c_m$.

The principal costs are the multiplications for verifying membership in $\{0,1\}_{GF(q)}$ and the executions of verifiable secret sharing. Note that all the verifications of membership are performed simultaneously, so the messages from the different executions can be bundled together. The same is true for the verifications in the VSS. The total cost of the scheme is $\Theta(n)$ multiplications and additions in shares, which can be all done in a constant number of rounds.

5 Conclusion

Two areas of research are tied in this work: the study of privacy-preserving statistical databases and that of cryptographic protocols. It was inspired by the combination of the computational power of the noisy sums primitive in the first area and the simplicity of secure evaluation of sums in the second area. The effect is to remove the assumption of a trusted collector of data, allowing individuals control over the handling of their own information.

6 References

1. C. Dwork and K. Kenthapadi. Our Data, Ourselves: Privacy via Distributed Noise Generation

2. D.J.C. MacKay. Information Theory, Inference, and Learning Algorithms

3. C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calirating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Theory of Cryptography Conference*, pages 265-284, 2006

4. A. Gabizon, R. Raz, R. Shaltiel. Deterministic extractors for bit-fixing sources by obtaining an independent seed. In Proceedings of the 45th IEEE Symposium on Foundations of Computer Science, pages 394403, 2004.

5. D. Freedman, R. Purves, R. Pisani. Statistics