Anonymity and Severity Analysis for Data Leakage Detection

Jay Velasco

Fordham University: Cybersecurity

*Abstract*— The number of records that were compromised often measures the severity of a data breach. Organizations may monitor for data leakages and provide alerts on certain criteria. Specifying alert criteria can help a company better utilize its time and resources. Providing data models to measure the severity of leakages can aid in the detection process. Severity can be based on many factors and should consider how much an attacker can infer about a subject from the leaked data. This work proposes an extension of L-Severity called KL-Severity. KL-Severity improves the accuracy and flexibility of L-Severity by considering well-represented sensitive values and other attribute classifications. An analysis of the used privacy metrics in this research was conducted. *(Abstract)*

Keywords-Data Leakage Detection, Privacy Enhancing Technology

# Introduction

Ponemon Institute performed a study in 2016 involving 383 companies from 12 different countries. The research found that there has been a 29% increase in in the total cost of a data breach, reaching an average of $4 million. Each record has an average cost of $158. The healthcare industry had the highest cost of $355 per record. [10] Data is growing at a rapid pace, .5% of all data is analyzed and this amount is decreasing. [13] To address the growing data, technology is changing and more investments in modern data infrastructure are being made. The investments are to improve data analytics, which includes real-time data processing and visualization. As data increases and technology advances, so will the variety and severity of attacks. Privacy and sensitivity of all attributes will contribute to the severity of a data breach. The cost for an organization to be prepared for a breach is fixed. The increase per person that an organization spends on security has gone up 15% since 2013. [10] The cost can be attributed to investments in resources and Data Leakage Prevention (DLP) technologies. The quicker an organization can respond to a breach will reduce its negative impacts. Stronger data governance, hiring a CISO, having an incident response and business continuity plan can help detect and mitigate data breaches. Leakages caused by cyber criminals are more expensive and harder to detect than those caused by system or human errors.

Regulations have been increased to protect the confidentiality of users. The European Union Agency for Network and Information Security (ENISA) handles information and network security throughout the European Union (EU). Guidelines set by ENISA are to Prepare, Detect, Notify and Respond to a security incident. [14] Accuracy and severity must be measured when handling confidential user information. Data must remain anonymous and privacy preserving techniques should be applied. Obtaining user consent when handling sensitive data may be required. Data handling must be done with intent to fulfill a purpose.

Many variables are involved when measuring the severity of a data breach. The criticality of the data can be determined on sector-based analysis. For example, a breach of confidential data of an organization can negatively impact their stock price. [5] Goel et al. saw a 5% decrease in stock price when a company is a victim of a confidential data breach. Data breaches not involving confidential information had no effect on the stock price. Organizations lost 2.1% of their market value within a timespan of 2 days from disclosure. [5]

The number of records leaked and the cost of a breach have a positive correlation. However, the severity of what was leaked may vary. For example, the disclosure of a specific disease may impact the life of an individual worse than others if disclosed. Those with expertise within their industry must define the labeling of attributes for a given domain. Vavilis et al. created data models with certain assumptions such as a disease, HIV, can have a major impact on the life of a subject if the data was disclosed. The severity of the disease increased as well as its medication to treat it. The medication received a high severity score because it can be used to infer the disease of a subject. The impact of a data leakage on an individual’s life is assumed to outweigh the number of records that was leaked when measuring severity in this research. For example, 10 records that reveal patients have the cold virus may be considered less severe than 5 records that reveal patients with HIV. Existing conditions when a leak occurred may also impact severity. Records leaked maliciously can remain undetected longer, which increases the negative impact on an organization. Other factors that increase severity are linkages and the frequency of an attack. Linkages are the relationships that are publically available that can be used to reveal sensitive information. This research analyzes the impact of different privacy metrics on the severity of a data leakage. KL-Severity, a model for quantifying severity weighing different data classifications with emphasis on well-represented diversity is proposed.

# Previous Work

## K-Anonymity

Sweeney et al. proposed K-Anonymity, which requires quasi-identifier values occur at least k number of times within a Q-Block. A q-block is a grouping of tuples that will have the same values for their quasi-identifier attributes. Quasi-identifier attributes can be used in combination to reveal a unique entity.87% of individuals can be identified by their 5-digit zip code, gender and date of birth. [3]K-anonymity protects against inference and linkage attacks. Sensitive attributes can be breached through unintended disclosures. Data that is retrieved in a single query may not violate the K-Anonymity rule. However, when the data is combined with other queries, it may reveal sensitive attributes. Generalized data can be released and unintentionally disclose information about individuals. Inference attacks involves linking attributes to other data sources. The government and medical industries commonly release information containing attributes unaware of other related data sources. Security of data can protect against a direct data breach, but not from information leaked through inference.

Attacks and vulnerabilities on privacy are not new within the security community. For example, statistical databases are released to provide data for research in data mining or fraud detection. A technique to generalize the data involves adding noise, which can damage the integrity of the information. Multi-level databases (MDB) store data in different classifications. Data is divided into higher and lower classified information. A vulnerability that can be found in a multi-level database is when lower classified information is used to infer higher classified records. A way to mitigate this vulnerability is through architecting a strong database design. However, the replication of data after a generalized table is released cannot be controlled. Data that leaves the original source can be copied and manipulated many times after. There is little to no oversight of handling the data once the data has reached multiple receivers. To avoid this vulnerability, all sensitive data can be suppressed, but this technique may decrease the utility of the data.

K-Anonymity is susceptible to different attacks. An unsorted matching attack occurs when positions of the tuples in each generalized table match the private table. To prevent this attack, randomly sorting the data can be done. Complementary released table vulnerability is when two generalized tables form a linked table. The linked table is used to combine quasi-identifier values to uniquely identify rows in the private table. The complementary released table vulnerability can be addressed by using the quasi-identifiers of the original table. Another technique is to base the new generalized table after the original table that was released. When basing the new generalized table from the original table, no value should be more specific than the values in the original table. For example, if the original table generalized their zip codes to 0213\*, the new table should not be more specific with 02139. Another vulnerability is a temporal attack. Temporal attacks occur when new data is added to the private table over time and a new generalized table is released. Linking the original released table with the newly released table can reveal unique rows. A way to avoid this is to base the newly released table on the original released table including the newly added information.

## L-Diversity

Ashwin Machanavajjhala et al. presented two attacks on K-Anonymity, homogeneity and background information attacks. An attacker can discover sensitive attributes when the data is not diverse enough. A homogeneity attack leaks information due to the lack of diversity in the sensitive attribute. An attacker may have background knowledge, which can be used to infer sensitive attributes. Ashwin Machanavajjhala et al. proposed Bayes-Optimal and L-Diversity. Bayes optimal is an algorithm that works under the assumption that the data publisher and adversary know the complete distribution set of sensitive and non-sensitive attributes. L-diversity provides privacy without the data publisher knowing how much background information an adversary may have. Although Bayes optimal covers a wider scope, it is not practical in use. It is unlikely that the adversary and data publisher have the complete knowledge of the sets of sensitive and non-sensitive attributes.

Each block of quasi-identifier groups, or q-blocks, should have at least *l* frequency of sensitive attributes. The frequency of sensitive attributes can protect against knowledge an attacker may know. Ashwin Machanavajjhala et al. proposes two algorithms to define well representation of sensitive values called Entropy and Recursive L-Diversity. Entropy diversity ensures that each q-block has well represented groups of sensitive attributes. The more uniform a q-block is, the higher the entropy. Recursive diversity is an algorithm that measures the frequency, but is implemented differently. Ashwin Machanavajjhala et al. proposes other algorithms to handle non-sensitive attributes, which involve variations of entropy and recursive diversity.

A q-block is considered (c, 2)-diverse if the frequency of the most occurring sensitive value is less the product of c and the sum of the remaining frequencies. Let *r* represent the frequency of an attribute.

## L-Severity

The M-Score calculates a severity metric, but has limitations. For example, to calculate the M-Score a Raw Record Score (RRS) is needed. The RRS has a maximum of 1 and the row with the highest RRS is used as the Final Record Score (RS). The RS is then used to derive the M-Score. In order to calculate the RS, there is a Distinguishing Factor (DF) that the RRS is multiplied by. Although not explicitly stated in the paper, DF is set to a constant .5. The M-Score was then calculated against 3 different cases. Case 1.x[[1]](#footnote-1) exposed the min and max limitations and Case 3.x shows although the leaked table in case 3.2 had less records, the diseases overall were more severe. To accommodate this, L-Severity was proposed. L-Severity will aggregate the node sensitivity of each sensitive attribute per row.

**Raw Record Score:** S is the set of sensitive attributes.

**Distinguishing Factor:**  Is the number of quasi-attributes within the row. Quasi attributes are attributes that are pre-defined and can be used to identify an entity by linking it to other sources.

**Final Record Score:**  Represents the source table.

**M-Score:**  Represents the number of leaked records. Variable x is defined by an analyst and influences the impact of the number of rows that was leaked.

**Record Sensitivity:** NS Represents the Node Sensitivity that is defined in the domain’s data model.

**L-Severity:** For each leaked row, aggregate the record sensitivity.

Table 1 Score Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | L-Severity | M-Score | | |
|
| *x*=1 | *x*=10 | *x*=100 |
|
| Case 1.1 Case 1.1 | 1.700 2.900 | 2.000 2.000 |  |  |
|
| Case 2.1 Case 2.2 | 1.150 2.100 | 2.000 2.000 |
|
| Case 3.1 Case 3.2 | 1.950 2.900 | 3.500 2.000 | 0.607 0.574 | 0.509 0.507 |
|

Table 1 shows the result matrix of M-Score against L-Severity with different values of x. X is only applicable to M-Score. Case 3.1 and 3.2 have different results, L-Severity scores the table in case 3.2 having a higher sensitivity score than what was given in M-Score, .507. Therefore, L-Severity takes account the severity of the entire table and is not limited by the min or max values in M-Score.

## M-score

The M-Score requires sensitivity functions to be defined by domain experts. M-Score was developed to provide a measurement of misuse. Harel et al. describes four dimensions of what they refer to as misuseability; number of entities, anonymity, number of properties and their values.

# KL-Severity

K-Anonymity and L-Diversity provide rules to prevent disclosing any sensitive information and to the best of our knowledge, has not been included in any published severity research. L-Severity uses a distinguishing factor, but does not evaluate out privacy preserving metrics on calculating the severity. L-Severity applies sensitivity scores only to sensitive attributes. We propose KL-Severity that expands on the L-Severity model to include different types of data classifications and to attach a weight to each set. KL-Severity also considers the frequency and number of distinct sensitive attributes in a table. The weight attached to a classification set will allow for more detailed analysis. For example, if we want to place stronger emphasis on the quasi-identifier attributes, we can increase the weight.

Quasi-identifier values can impact the severity of a data leakage. For example, a leakage involving diseases can have different impacts on individuals at certain stages of their lives. Disclosure of a disease can have varying consequences. LT1 and LT2 are two tables with identical attributes except for age. Assuming the severity score for the diseases are equal, the traditional L-Severity model would conclude that these two tables that were leaked have equal severity. However, the severity of LT2 can be debated because the table contains sensitive information regarding a minor.

LT1

|  |  |  |
| --- | --- | --- |
| Job­ | Age | Disease |
| Student | 100 | HIV |

LT2

|  |  |  |
| --- | --- | --- |
| Job | Age | Disease |
| Student | 10 | Ebola |

Quasi-Identifier Sensitivity Scores

|  |
| --- |
| Quasi-Identifier |
| f(age < 18) = .8 f(age) = .5 |

Using the L-S­­everity model only sensitive attributes are considered without any separation from quasi-identifiers or non-sensitive attributes. Non-sensitive attributes may be misclassified and can give more information than intended. Therefore, attaching a constant score for any added attribute can be beneficial when detecting for data leaks. A limitation of scoring severity is that the domain of these attributes and the classifications must be maintained. If an attribute is misclassified an attac­­­­­­­­ker can target non-sensitive attributes to prevent detection.

**JSON Example**

{ property\_1: value\_1 },

{ property\_1: value\_1, property\_2: value\_2 }

JavaScript Object Notation (JSON) is a data exchange format, which allows for easing parsing. JSON can come in different formats. An object’s properties can be parsed and properties can be added without the receiver’s knowledge. If the scope of the user’s validation does not include checking the properties, this data may make it through into their system or forwarded to another party. This unchecked property can go undetected. If severity is being tracked and the property is not in a maintained score table, an adversary can leverage this to pass or receive sensitive information. Giving a constant score to non-sensitive attributes can be useful when reading unstructured data.

LT3

|  |  |  |
| --- | --- | --- |
| Job | Age | Disease |
| Student | \* | HIV |
| Student | \* | HIV |
| Student | \* | Ebola |
| Student | \* | HIV |

LT4

|  |  |  |
| --- | --- | --- |
| Job | Age | Disease |
| Student | \* | HIV |
| Student | \* | Ebola |
| Student | \* | Cancer |
| Student | \* | Meningitis |

LT3 and LT4 are examples of two leaked tables that we assume have the same severity scores for their sensitive attributes. The difference between LT3 and LT4 is the number of distinct sensitive attributes that are disclosed. LT3 contains 2 distinct values for disease while LT4 has 4. An inference attack on LT3 would be more successful in performing than LT4 because of the lack of diversity in the sensitive attributes. For example, if Alice knew Bob did not have Ebola and knew that Bob attended Hospital A where LT3 was leaked. Alice can infer that Bob has HIV. If LT4 was leaked from Hospital A, Alice still knows that Bob does not have Ebola. However, Alice will have to deduce amongst three other diseases – Cancer, HIV and Meningitis. The scenario described can be quantified using KL-Severity.

**Definition 1:** Let L( be the rows of the leaked table.

**Definition 2:** Let represent the number of leaked rows. DF’s definition will be borrowed from L-Severity.

**Definition 3:** Let be the Classification Sensitivity of a row. For each classification of a row, record the node sensitivity of the attribute. Let be the weight given to the classification.

**Definition 4:** Let represent a q-block that row *r* is a party of.

**Definition 5**: The DivFactor, or diversity factor, represents the number of distinct sensitive values for a given q-block. Let QBS represent the set of sensitive attributes that reside in the q-block.

**Definition 6:** Let WR represent the added weight if a q-block is not well represented.

Given the above definitions LT3 would have a higher severity than LT4. We can assume *c* = 1 and the severity score of all the diseases are the same.

Case 1: Severity of LT3 and LT4

|  |  |  |
| --- | --- | --- |
|  | LT3 | LT4 |
| Equation | 4 \* (20 / 4 / 2) + 1 | 4 \* (20 / 4 / 4) + 0 |
| Severity Score | 11 | 5 |

# Discussion

The comparison of the impact of the Dependency Factor (DF) in L-Severity was done against K-Anonymity. This research did not find a significant impact on the severity when alternating algorithms. For example, if a table is conforming to the K-Anonymity rule, the DF can also remain constant or decrease severity. Having a higher DF metric will reduce the severity of a row. However, a higher DF score does not guarantee that a leaked table conforms to the K-Anonymity rule. In order for a record to follow the K*-*Anonymity rule, it must be part of a group of records that is at least *k* in size. After our analysis of M-Score’s DF metric and K-Anonymity we concluded that the metrics will have equivalent influence on the severity score. An addition that can be added to the overall table is a weight that determines if a table followed the K-Anonymity rule. K-Anonymity is a good baseline for measuring privacy within a generalized dataset. We attempted to measure the impact of considering how far off a group of records were from *k.* The farther the number of unique quasi-identifiers a group of records is from *k,* the higher the severity will be. This correlates with a lower DF metric. For example, if there is only 1 distinguished record out of *n* records, *n/1* is greater than *n/k.* Assuming that *k* is larger than 1.

Having the capability to attach weights to different classifications allows for more detailed analysis. For example, it is possible to weigh privacy higher than other classifications. Classifications can include the sensitive, quasi-identifier, non-sensitive attributes or attributes that are specific to an industry. An example a classification that can be added that is industry specific would be *value*. The addition of measuring the diversity and frequency of the sensitive attributes contributes to the accuracy of the severity score. As seen in Case 1 under Section III, the severity a table is increased when the sensitive attributes are not diversified and well represented. The non-sensitive attributes can be represented in a separate classification.

## System Architecture

Normal traffic of an application or system must be defined. Vavilis et al. use examples where data is queried from a system. A baseline of “normal” should be defined in order to detect anomalies in severity. DLP technology is correlated with organizations that have teams to prevent data theft. [11] McAfee performed a study where 64% of security professionals within firms that experienced a data breach agreed that the breach could have been prevented if their firm used DLP technology. DLP technology is a top tool in detecting insider threats. A tool that detects the severity of data being retrieved can be helpful when investigating security events. Future work will involve architecting a system that can be used to measure severity on data sets. Finding vulnerabilities in detecting KL-Severity can be done. An interesting test would be to attempt to bypass security controls by retrieving sensitive attributes while keeping within “normal” severity measures. Retrieving the sensitive data in smaller chunks and at higher frequencies can be an attack to perform on KL-Severity.

KL-Severity Architecture

The KL-Severity Architecture proposes a way to consolidate data from multiple unrelated sources in one application. The proposed architecture is beneficial for scaling applications. Implementing severity measurements at a database level does not consider data from other sources unless database links are set up or data from other sources is moved to the targeted database. However, the latter suggestions can only be done if the involve parties are in agreement with each other with sharing data. The sharing of data may not be possible between involved parties and would have to be consolidated in other ways. Vavilis et al. proposes a data model in a hierarchical format. Traditional data structures such as binary and balanced trees can represent the data model’s relationships better than what may be available in database technologies.

# Conclusion

The importance of data leakage prevention is relevant in today’s media and influences how we use and ingest data on a day-to-day basis. Previous work shows an emphasis on finding a severity metric that takes account the entire table. However, the result can be impacted by privacy metrics such as the distinguishing factor. Providing security metrics at a database level is beneficial, but having the option to do so at an application level can be more robust.

Tracking transactions within an application can alert an organization at the time a possible breach has occurred. A breach may go undetected until a victim reports a problem or an attacker advertises the data on the black market. ADLP may be tracking when sensitive data is released or based on a measurement of severity, such as KL-Severity. Severity is complex and can involve many dimensions that are shared or specific to an industry. However, there are few a publications that focus strictly on the severity of only the data and the definitions of the impact of values for sensitive, quasi-identifier and non-sensitive attributes. Vavilis et al. created a model to quantify severity by attaching severity scores to values within a sensitive domain. L-Severity does not separate different classifications of the data. We propose KL-Severity that is a more scalable and accurate model.

Data can be retrieved in different formats from various sources. For example, an API can accept or send JSON that may have varying properties. Due to the unstructured format that the data can come in, unexpected attributes may be passed. Depending on how this data is used, extra information may be leaked or accidentally disclosed to an unauthorized user. An example could be a data dump of values that need to be emailed to another group or data that is passed into another system. This can cause system errors, rejections and leakages.

Future research can involve more work on data classification. Ashwin Machanavajjhala et al. proposed different algorithms to handle *don’t care sets* (DCS). DCS represents sensitive attributes that have no effect on severity. Although DCS has no effect on severity, when mishandled can reveal sensitive attributes. Other metrics, such as T-Closeness, are encouraged to be evaluated using KL-Severity. We plan to release future work on this topic involving other privacy metrics. A system that will be used in an experiment with real data and testing various security events will be created in the future.

# References

1. Machanavajjhala, Ashwin, et al. "l-diversity: Privacy beyond k-anonymity." ACM Transactions on Knowledge Discovery from Data (TKDD) 1.1 (2007): 3.
2. Vavilis, Sokratis, Milan Petković, and Nicola Zannone. "A severity-based quantification of data leakages in database systems." Journal of Computer Security 24.3 (2016): 321-345.
3. Sweeney, Latanya. "k-anonymity: A model for protecting privacy." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10.05 (2002): 557-570.
4. Vavilis, Sokratis, Milan Petković, and Nicola Zannone. "Data leakage quantification." IFIP Annual Conference on Data and Applications Security and Privacy. Springer Berlin Heidelberg, 2014.
5. Goel, Sanjay, Christopher Brown, and Hany Shawky. "Measuring the impact of security breaches on stock valuations of firms." Information & Management 46.7 (2009): 404-410.
6. Ponemon Institute. (2016). 2016 Cost of Data Breach Study: Global Analysis. The Ponemon Institute.
7. Wagner, Isabel, and David Eckhoff. "Technical privacy metrics: a systematic survey." *arXiv preprint arXiv:1512.00327* (2015).
8. Harel, Amir, et al. "M-score: A misuseability weight measure." *IEEE Transactions on Dependable and Secure Computing* 9.3 (2012): 414-428.
9. Only 0.5% of All Data is Currently Analyzed, A.R. Guess, June 10, 2015
10. Recommendations for a methodology of the assessment of severity of personal data breaches, Data Protection Authorities of Greece and Germany, Clara Galan Manso, ENISA, Sławomir Górniak, ENISA, December 20, 2013
11. Grand Theft Data - Data exfiltration study: Actors, tactics, and detection, McAfee, 2015

1. X represents cases 1 and 2 [↑](#footnote-ref-1)