IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 31, 2022, accepted March 7, 2022, date of publication March 11, 2022, date of current version April 1, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3158934

Hash-Comb: A Hierarchical Distance-Preserving Multi-Hash Data Representation for Collaborative Analytics

ABDELRAHMAN ALMAHMOUD[1], (Member, IEEE), ERNESTO DAMIANI[2,3], (Senior Member, IEEE), AND HADI OTROK[2], (Senior Member, IEEE)

[1]Technology Innovation Institute (TII), Abu Dhabi, United Arab Emirates
[2]Center of Cyber-Physical Systems (C2PS), Department of EECS, Khalifa University, Abu Dhabi, United Arab Emirates
[3]Emirates ICT Innovation Center (EBTIC), Khalifa University, Abu Dhabi, United Arab Emirates

Corresponding author: Abdelrahman Almahmoud (abdelrahman.almahmoud@tii.ae)

ABSTRACT Data privacy regulations like the EU GDPR allow the use of hashing techniques to anonymize data that may contain personal information. However, cryptographic hashing is well-known to destroy any possibility of performing analytics. Homomorphic crypto-systems allow computing analytics over encrypted data, but cannot guarantee privacy compliance without being coupled with specific privacy-preservation provisions. In this work, we present a novel distance-preserving hashing scheme supporting both regulatory compliance and collaborative analytics. Our scheme achieves regulatory compliance by relying on standard cryptographic hashes while preserving a controllable notion of distance between data points.

INDEX TERMS Data privacy, distance-preserving hashing, big data, homomorphic encryption, hashing, quantization kit.

I. INTRODUCTION

In recent years, organizations and the general public have become increasingly wary of how their data is being collected and shared [1]. To protect personal data, several regulations covering various aspects of data privacy have been introduced, most notably the European Union's GDPR, China's draft Personal Information Protection Law,[1] and the Dubai data law, which governs the sharing of data at the city level. All these regulations encourage protecting user's privacy through the use of certified implementations of proven hashing and encryption techniques. Generally speaking, cloud providers using untried algorithms to anonymize their customers' data may fail to achieve regulatory compliance, facing substantial fines and sanctions, regardless of the algorithms' actual effectiveness [2], [3] [4], [5]. The widespread adoption of Machine Learning (ML)-as-a-service has increased disclosure risk [6], as access to ML models' parameters can lead to disclosing the data sets used to train them. *Homomorphic Encryption* (HE) has been proposed since long to perform computations over encrypted data, protecting ML data assets during training and inference [7]. Our work is motivated by widespread concerns that HE encryption techniques may not guarantee that encrypted personal data can be considered as anonymous data, and therefore processed outside the scope of the data protection regulations like the EU's GDPR[2] [2]. Three relevant factors must be considered when assessing the level of security of encrypted data: the strength of the encryption algorithm used, the length of the encryption key, and the security of encryption key management [10]. If an organization holds some third party's data (acting, in GDPR terms, as the data *controller*) in encrypted form but does not hold nor can access the decryption key in any way, it is reasonable to assume that it will not be able to access any personal information within the data [5]. In this case, the *controller* can safely regard the data as anonymous and outside the scope of the GDPR. The strength of the

The associate editor coordinating the review of this manuscript and approving it for publication was Gaurav Somani.

[1]On April 29th, 2021, China issued the second version of its Personal Information Protection Law (''Draft PIPL''), which is currently in the public consultation rounds.

[2]Differential privacy techniques based on noise addition [8] also allow data owners to add noise to their data according to a privacy-vs-accuracy budget, but this budget does not directly reflect the amount of information released after the noise model is applied [9], and is therefore unfit for achieving regulatory compliance. Therefore, we do not consider them in this paper.

cryptosystem and the assumed separation between encrypted data and the decryption key, however, need to be verified, following international standards like NIST FIPS 140-2.[3] Currently, no HE implementation has been validated according to FIPS 140-2 [11], and homomorphically encrypted personal data cannot be treated as anonymous information falling outside the scope of EU GDPR. Of course, HE could be used in association with verified privacy-preservation techniques like secure multi-party computation and aggregation [12]. Coupling HE and secure multi-party computation, however, requires a significant amount of customization to handle heterogeneous data types [13]. Cryptographic hashing is a popular way of anonymizing personal data, as it does not require verifying any key management scheme. The use of hashes for data anonymization has been encouraged by international security agencies. The European Network and Information Security Agency (ENISA) has released recommendations for the GDPR-compliant use of hashing for anonymization purposes[4] and hash functions to be used for data anonymization have been subject to international standardization.[5] Today, several tool-kits are available for inter-organizational hashed data comparison [14]. Such tools support collaborating organizations in answering simple questions like ''how many customers do we share?'' while keeping the data anonymous. We argue that there would be much more value in allowing data owners (or external services) to seamlessly combine anonymous data to carry out more advanced data analysis, involving arithmetic and distance computation. In the aftermath of the COVID pandemics, collaborative analysis-as-a service on anonymous healthcare data is a very appealing prospect for cloud providers [15]. However, for data owners to take part in collaborative analysis, data must be anonymized using techniques whose implementations have been approved by international standards, preventing any regulatory hazard. Unfortunately, approved hashing techniques do not support arithmetic, limiting the value of hashed data for collaborative analysis. In this paper, we address the challenge of supporting collaborative, privacy-preserving data analysis while using approved hash functions. To this end, we propose a novel multi-hash representation called *Hash-comb*, which represents data as a hierarchical (multi-level) hash generated by quantization of data values using multiple granularity levels. We validate the performance of our Hash-combs using two data sets related to online advertisements that cover different data types (numerical sensor data and IP network data) and a total of 84 test scenarios.

II. BACKGROUND

Due to the large body of work surrounding data hashing, in this Section we will review only hash techniques that show some potential for supporting computations on hashed data. We will use the term *hash function* to designate any a process which transforms a plain-text data input into a fixed length character series, regardless of the size of input data. The output is called hash value or digest; for brevity, the term ''hash'' will sometimes be used both in reference to the hash function as to the hash value, which is the output of playing this function on a particular data input. We will refer to the set formed by all possible inputs as the *data domain* or *data space*, while the set of all possible hash values will be called *hash space*. Standard hash functions approved for data anonymization are purposefully designed to cause an *Avalanche Effect* [16], ensuring that changing a single bit in the input will result in a substantial change in the corresponding hash value. In the remainder of the section we review some alternative hashing techniques that organize the hash space by preserving some information about the data space distance.

A. DISTANCE-PRESERVING HASHES

Distance-Preserving Hashes (DPHs) differ from classic hashes inasmuch they map input values that are close to each other in the data space into neighbours in the hash space. Distance-preserving hashes can be defined for also categorical data: a recent work [17] describes how text hashes can preserve similarity between documents.

Data-dePendent Hashes (DPHs) are a family of hash functions whose hash values depend not only on the inputs, but also on other points in the data set. Quantization-based DPHs include an additional quantization step before hashing the data. The quantization step facilitates both compression and distance preservation. Context-triggered approaches are DPHs that rely on the occurrence of *trigger values* within the data. The time-honored *Context Triggered Piecewise Hashing* (CTPH) [18] is still one of the most popular similarity-preserving hashing schemes to date. The algorithm uses a combination of *rolling hashing*, i.e hashing based on a dynamic window moving over the input data flow k bits at a time, and traditional hash functions which is applied to fixed size data chunks when the rolling hash produces an output that matches a pre-defined trigger value [19]. Multi-Resolution Similarity Hashes (MRSH) [20] and MRSH-v2 [21] are variants of context-triggered hashes that use multiple triggers to reduce the complexity of choosing a suitable trigger value. Additional improvements were introduced in [22] by coding the hash into a hierarchical Bloom filter to speed up hash comparisons. bbHash [23] compares its inputs to randomized data blocks rather than trying to rely solely on the inputs to compute the hash. Initially, a sequence of pseudo-random data blocks are generated and are compared to the input. If the Hamming distance between the input and the random block is smaller than a certain value, the output is adjusted accordingly, otherwise no changes occur. It is noteworthy that the size of the final hash is not fixed, which could have negative implications on hashed data indexing. fbHash [24] comes in two variations, called

[3]Approved encryption methods are documented in the Annex A of FIPS 140-2.

[4]https://www.enisa.europa.eu/publications/recommendations-on-shaping-technology-according-to-gdpr-provisions

[5]A list is available a part of NIST FIPS PUB 180-4 Secure Hash Standard (SHS).

FbHash-B and FbHash-S. Both variations are *rolling hashes* the former is designed for byte-level similarity detection while the latter is modified for semantic matching.

N-gram-based hash functions rely on n-element subsets of the input domain. Typically, they split the data into n-grams, and then capture the n-gram histogram or some variations of it. Nilsimsa [25] is an n-gram based distance-preserving hash which is designed to detect similarity between e-mails. It uses a five character sliding window which captures trigram combinations and stores them into an accumulator. N-gram hashing proved effective in preserving e-mail similarity against known spamming modification techniques, and is still popular in network traffic classification applications [26]. Trend Micro Locality Sensitive Hash (TLSH) follows the n-gram structure of Nilsimsa but treats inputs as bytes rather than characters. Furthermore, it uses quartiles instead of the median when constructing the digest in order to capture homology in binary data and images [27], [28]. N-gram hashes have been successfully used in applications such as forensics, spam detection, databases and image retrieval which require searching over large data sets for similar files. MvHash-B [29] is a basic quantization based DPH with low computation time. The core principle is to partition a binary input into equal sized chunks. Then, the algorithm runs a majority vote over the chunks to obtain the dominant bit (0 or 1). This allows for the transformation of the input into a compressed sequences of identical bits. The newly obtained file is then processed using *run-length encoding* to compute the final hashed value.

A common trait of all the DPH schemes reviewed above is providing some degree of distance preservation. However, computing this approximation can be a major bottleneck in Big Data applications [30]. Literature on Fast Forensic Similarity Search (F2S2) [31] argues that trying to find similar files using distance-preserving hashes over large amounts of data is very time consuming. The authors propose indexing digests based on the occurrences of certain trigrams in them. While F2S2 speeds up the process of finding similarities and is suitable for forensic applications, the platform must know which trigrams occur in the digest, potentially giving away some information about the contents of the plain-text, which makes it unsuitable for many applications. It is also important to remark that all hashes discussed so far capture some statically pre-defined distance. They are designed to work with certain types of data and capture a limited amount of information from the plain-text, making them hard to adaptable to different types of data or distances.

1) DATA-DEPENDENT HASHES

Data-dependent hashing techniques dynamically learn certain parameters from input data. Techniques that rely on learning hash codes gained popularity due to their success in Machine Learning applications. Recently, much focus has been given to representing the quantization step in distance-preserving hashes. An example of such work is [32] which argues that more attention should be given to the quantization step of hashing because it has a critical effect on the performance of the hash function. The work proposes Manhattan Quantization (MQ), which preserves the binary neighbourhood structure of the plain-text using Manhattan distance computed over the natural binary code (NBC). Another example of explicit quantization is Hamming Compatible Quantization (HCQ) [33], which formulates quantization as an optimization problem. The work defines and aims to minimize a distance error functions measured between data in the euclidean and hamming spaces. An example of these hashes is Discrete Graph Hashing (DGH) [34] which preserves the neighbourhood structure of the data set by building a graph based hashing model using anchor graphs. This model is also able to generate hashes for data out of the training set by minimizing the distance between a new point and its neighbours. A popular approach to data-dependent hashing is *deep hashing*, which utilizes neural networks to produce efficient hashes. An example is Text Hashing with Convolutional neural networks (THC) [35]. THC trains a Convolutional Neural Network (CNN) for generating distance-preserving hashes of text data. The data set is prepared by embedding the words using existing, pre-trained word vectors; then, explicit distance information and features are attached to them to obtain the final hash. We argue that current learning-based hashing techniques are unsuitable for general use, as they must be tailored to specific data types. Also, they are unsuitable for collaborative analytics scenarios as they require some data in plain-text to be available for the hashing function to learn its parameters.

III. HASH-COMB

Hash-Comb is a multi-hash data representation scheme supporting computations on hashed data. Hash-Combs rely on standard-approved cryptography hashes as building blocks and enable distance computation on hashed data with the granularity needed to fit the analytics' privacy requirements and accuracy constraints. Our scheme works with different data types, is simple to implement and complies to data sharing regulations that require the use of specific cryptography techniques [36]. Hash-Comb multi-hash representation consists of an array of (ω) cryptographic hashes, which - as a whole -captures the distance information from the corresponding plain-text data item with ω granularity. Individual hashes within Hash-Combs are referred to throughout this paper as the ''dimensions'' of the data representation. Unlike distance-dependant hashing schemes, we do not directly encode distance information within our multi-hash representation. Rather, we introduce a data preprocessing stage (*Quantization Kit*). The basic working principle of the Quantization Kit is to split the data space into (C) regions. The configuration of C is tweaked according to two parameters, γ and R which represent the channel size and the size of data space respectively. This process is repeated for ω iterations with varying C, ω and R values. Each of the resulting ω dimensions covers a unique configuration of the C regions, resulting in a different granularity of distance

information to be encoded into the final hash. The example in Figure 1 demonstrates the process of splitting the plain-text space into a multitude of configurations visualized as layers. Each data point is represented using ω identifiers obtained from the channel it belongs to within a layer as described in Algorithm 1.

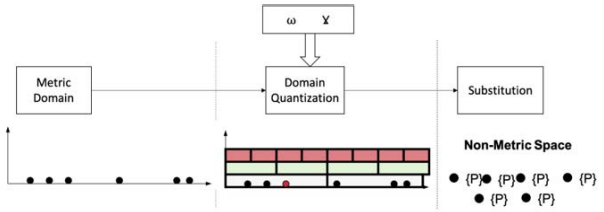


FIGURE 1. Generation of a Hash-Comb. Three quantizers capture the data at three different levels. The top level splits the data into two segments while the middle level splits it into four and the bottom level splits it into eight. In this example the resulting Hash-Comb contains three "dimensions"/elements: h_{a1}, h_{a2}, h_{a3} produced by each quantizer.

Algorithm 1: Generating a Hash-Comb

Input: R, C, ω
Output: $h_{0\ldots\omega}, C, \omega, \gamma, R$
for ω **do**
  $\gamma_i = \frac{\max R}{C_i} \parallel C_i = \frac{\max R}{\gamma_i}$;
  for *All* N **do**
    if $N_i \subset ch_i$ **then**
      $N_i \mapsto ch_i$;
    else
      Next;
    end
    $h_i = \text{Hash}(ch_i)$;
  end
end

For clarity, let us consider the example of a Hash-comb with $\omega = 3$ hashes shown in Figure 1. Each hash captures a different granularity of distance information from the plain-text data space, by partitioning it into three layers composed of $C = 2$, $C = 4$ and $C = 8$ channels respectively. After defining the regions of the plain-text data space, for each value in the data set, our scheme performs a cryptographic hashing process over each channel configuration, producing the hashes $[h_{a1}, h_{a2}, h_{a3}]$. This process allows for computing the (approximate) distance between two plain-text data points simply by comparing their Hash-combs.

A flowchart streamlining the process of generating a Hash-comb in a real life scenario is outlined in Figure 4, which details the execution of our hashing process. It involves a trusted environment (holding the plain-text data set) and a semi-trusted environment (the data collection/analysis facility). The workflow is executed as follows:

1) The first step is to choose a sub-sample from the plain-text which will serve to fine tune the parameters and obtain the desirable level of accuracy.
2) The data is then fed to the Quantization Kit, which is in charge of splitting the data space into regions according to the parameters and requirements. The output of this stage is a series of "dimensions" composing the building blocks for the Hash-Comb.
3) The output obtained from the Quantization Kit is then fed into an optional "Noise Function", which selectively adds noise into certain parts of the Hash-Comb's dimensions to add a layer of privacy.
4) The final step of generating a Hash-Comb is feeding the noisy data to the Hash Generator, which applies a cryptographic hash to every dimension, generating a series of hashed values.
5) The Hash-Combs are then collected and forwarded as a batch to the Semi-Trusted Environment.[6]
6) The Semi-Trusted Environment collects and stores batches of Hash-Combs from multiple participants via a secure line.
7) The collected Hash-Combs are then used to compute/train analytics (e.g, machine learning models) that require distance information.
8) Finally, the results are reported back to the data contributors and participants on another secure line.

A major advantage of Hash-Combs when compared to classic techniques [20], [29] is that our approach is not limited to Euclidean data spaces, where distances represent the shortest path between two points along a straight line. In fact, Hash-Combs are also suitable for approximating non-Euclidean *geodesic* distances [37], which have applications to network security [38], to tracing paths on 3D mesh objects [39], to clustering [40] and training Machine Learning models [41]. Revisiting the example in Figure 4, we can define a simple distance measure based on an element-wise equality check. Let us consider two distinct points a and b, each with a hash-comb h_{ai}, h_{bi} containing a set of three hashes $i = \{1, 2, 3\}$. The distance between the two hashes is computed by comparing each hash value of h_{ai} with the corresponding hash value from h_{bi}. This hierarchical comparison allows for bounding the distance between two points with γ as the max possible error.[7] Knowing quantization ranges as a prior, the data analytics can estimate the distance range between the two data points through the process of elimination. In case no match is found, the distance between the plain-text data points is estimated to be larger than the largest quantization interval, which makes the distance relation between the two points irrelevant in most cases. Having discussed applying our data representation to numerical data, we move on to discussing the use of Hash-Combs to represent structured data types of interest.

Many cyber-security applications benefit from merging Network/Netflow data from multiple data sources for

[6]It is possible to perform this step in real-time. However, for the sake of clarity, here we focus on batch collection.

[7]In our implementation, the hashes are organized by the quantization size, the first hash is the smallest and the space gradually increases until the last hash to optimize the comparison step.

TABLE 1. Notation table.

R	$\triangleq$	Range of the values in the dataset.
N_i	$\triangleq$	The dataset element at position i.
ω	$\triangleq$	Number of hashes.
γ	$\triangleq$	Channel size or channel interval / width.
C	$\triangleq$	Number of channels.
ch_i	$\triangleq$	Channel number.
h_i	$\triangleq$	A hash-comb where i is the hash index and $\omega \geq i \geq 0$.
k	$\triangleq$	Number of clusters.
r_k	$\triangleq$	The score of a plaintext cluster with the k_{th} anonymized cluster.

joint analysis. However, it is universally agreed that these types of data contain highly sensitive information. To illustrate the application of Hash-combs to Network / Netflow data type, we focus on the prefix distance in the domain of IP addresses because it has practical applications[8] in cybersecurity [42]. For our purposes, prefix distance is interesting because it requires a different segmentation strategy than previously described in Figure 1. Let us assume that we want to compute the prefix distance between the two IP addresses shown in Figure 2 using Hash-Combs with $\omega = 4$. Each dimension is then computed as shown in Figure 2: the first tooth of the Hash-comb is the result of hashing the first IP block, while the second and third are the results of hashing both the first two blocks and the first three respectively. The final tooth is the hash of the full IP address.[9] Computing the prefix distance between these two Hash-Combs reveals that the first three dimensions match each other, while the last one is different. This allows us to conclude that the prefix overlap is over a third of the IPs. It is possible to apply the same principle to define different ω and quantization sizes thereby narrowing the estimate further. Designing the Quantization Kit in a hierarchical manner rather than segmenting every IP block in isolation is a design choice discussed in our previous work [43].

A. DISTANCE-BASED ML USING HASH-COMBS

We now show how to use the Hash-combs prefix dimension distance to train a simple Artificial Neural Network model (ANN). Each dimension is encoded as an individual feature which is then fed individually into the ANN as a normalized numerical value. This *embedding process* is a simple translation from the hexadecimal hash value to a numerical representation. While it is true that this mapping introduces an ordering between dimensions where a distance could be computed between h_1 and h'_1, the avalanche effect of cryptographic hashing algorithm ensures that the hashed values of the same dimension are far enough from each other Figure 3. Thus, the analytics engine can perform dimension-wise comparison.

[8]For example, prefix distance can preserve *country neighborhoods*, which can be handy for grouping the bulk of botnet traffic.

[9]We could trivially increase the number of dimensions by segmenting the IP address into finer grains, which would result in a linear increase in size.

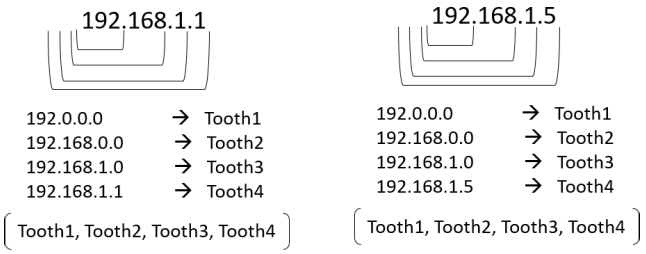


FIGURE 2. In this example, the quantizer captures the prefix distance in $\omega = 4$ stages. The first tooth captures the first block while substituting the remaining with 0. While the second tooth captures the first two blocks and the third captures the first three blocks. The final tooth captures the entire IP address.

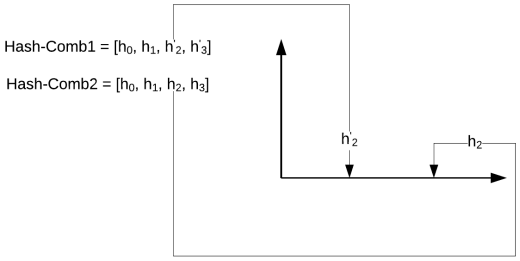


FIGURE 3. Embedding hash values into numerical space introduces an order between them. However, the diffusion property ensures that dimensions which are not equal are appropriately far from each other.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

For our validation, we used data sets collected for online advertisement fraud detection. Advertisers usually commission online advertisement companies to display their ads on partnering websites. When a user interacts with, clicks on or makes a purchase through these advertisements, the advertising company pays the website owner a fee for that interaction. Unfortunately, the size of the market and potential profit invites abuse and fraudulent behavior including infecting advertisements with malicious code and publishing misleading advertisements. However, the most common way of committing fraud is using bots to inflate click ad revenue (traffic sourcing) [44]. In this type of attack, a website owner utilizes bots to generate fake clicks and interactions with advertisements displayed on their own websites thus forcing advertising companies to pay them for clicks that never occurred in reality. Our experimental data set is based on a real data set of user clicks shared by an industrial partner. The data set contains the source (customer) and destination (site) IP addresses as well as various other information we do not describe for the sake of conciseness. In order to control bias in our experiments, we applied our technique to two versions of the data set. The first version is composed of >14M (14673922) records, extracted from raw data without further modification. The second is composed of around 170K (170396) records, selected to contain only unique source/destination IP pairs.[10] The rationale behind this second smaller version of the data set is making

[10]The significant size reduction is also due to removing a large portion of the data set.

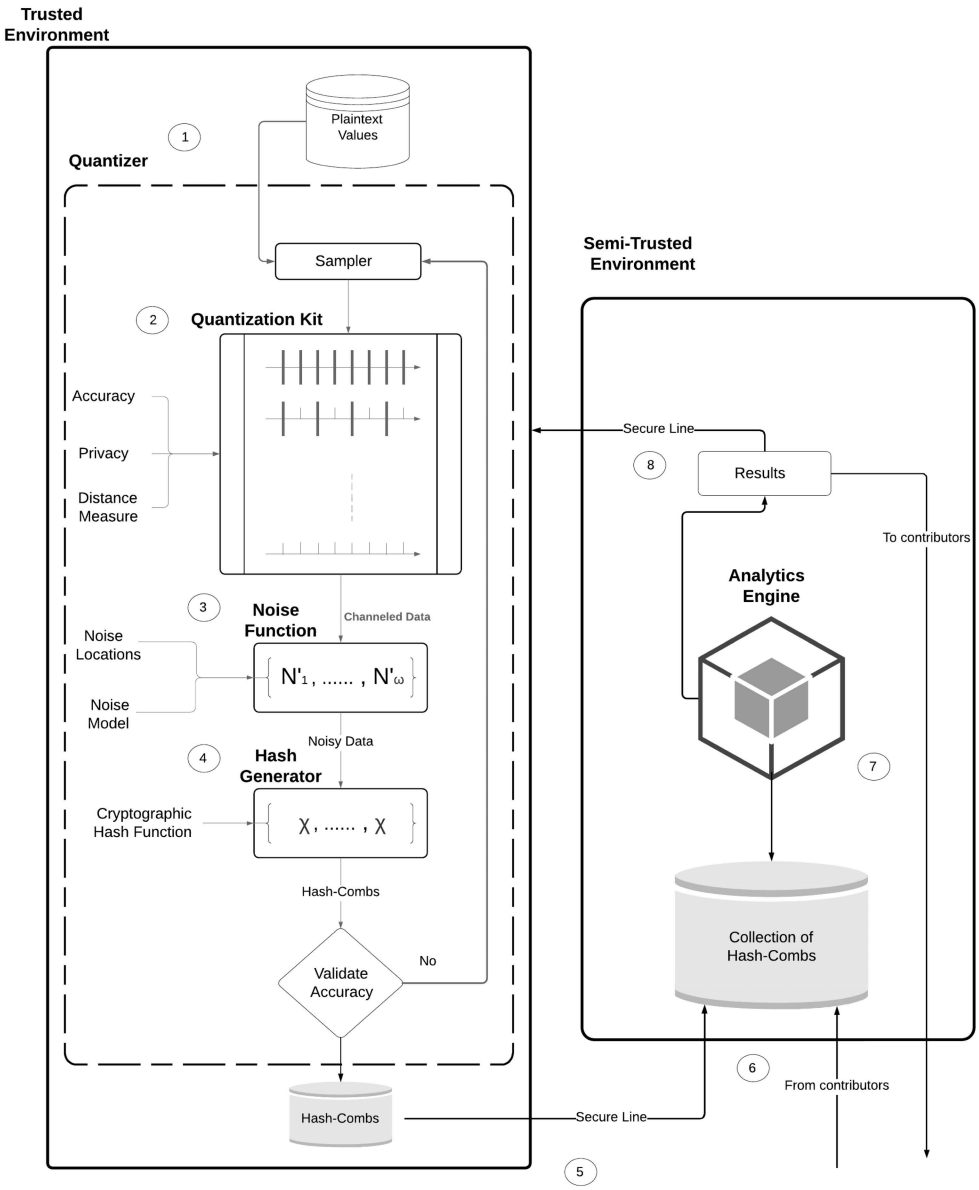


FIGURE 4. A workflow describing the proposed solution's deployment. The workflow is split into two parts, the first to be executed in a trusted environment (the data owner site) and the second in a semi-trusted platform at the data analysis facility.

plain-text data as diverse as possible to test our Quantizer's ability to group the IP addresses based on their prefix distance. Both data sets have been used to test the ability of our proposed hashing scheme in retaining distance in the hashed data space, in order to verify the performance of a classifier obtained by training the ANN with labeled Hash-Combs. The full details of our experimental setup are shown in Figure 5.

Being the experimental data sets unlabeled, our experiment (Figure 4) starts by automatically generating labels, using a modified version of the K-Means clustering algorithm.[11] The output cluster IDs are converted into labels attached to the Hash-Combs; then the labeled data set is used to train an Artificial Neural Network (ANN). The experiment is intended to verify that the ANN will be able to accurately map each Hash-Comb back to the cluster of the corresponding plain-text data point. To this end, the plain-text data is fed to the Quantizer, which generates multiple variations of Hash-Combs for experimentation purposes. Group 1 is set to $\omega = 4$ per IP address (total of 8 for source and destination) and group 2 is set to $\omega = 8$ per IP address (total of 16). Multiple data sets are created with a variety of ω values by removing dimensions gradually from both groups, as shown in Figure 4. Moreover, alternative configurations are also created, where dimensions are substituted with random noise. In the end,

[11]The clustering algorithm was amended to support computing the prefix distance between source and destination IP addresses as described in Section III.

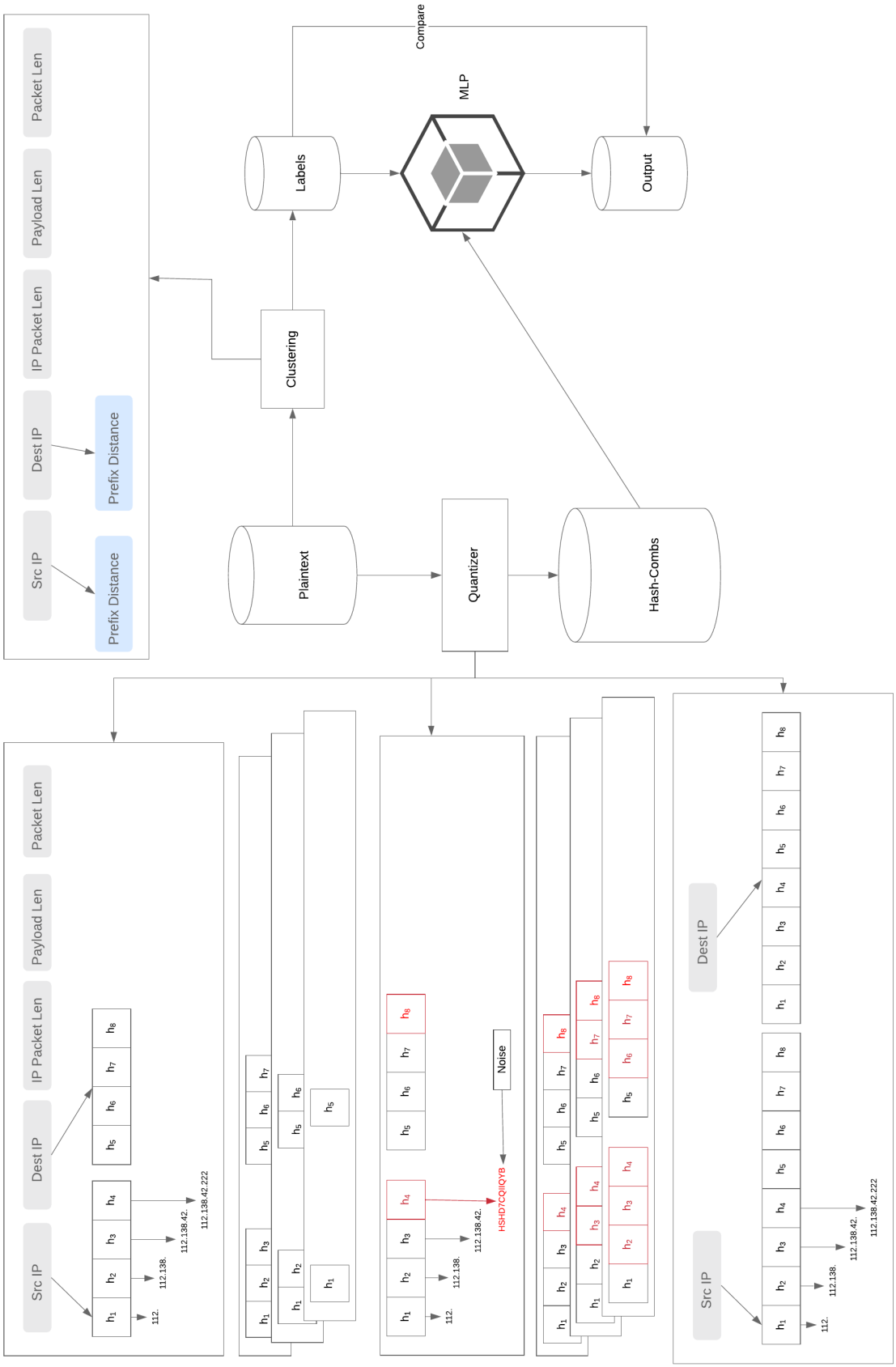


FIGURE 5. A synoptic view of the experiment showing all steps of data transformation.

we tested a total of 84 scenarios for both data sets. These configurations allow us to test the effects of adding more dimensions and the effects of manipulating them on the accuracy and performance of our scheme.

The ANN model was created using Tensorflow [45] and Keras. The model consists of an input layer (ReLU) which is adjusted based on the size of ω, four layers (ReLU) each composed of 1000 neurons which feed into two dense layers (ReLU) each consisting of 500 neurons. A dropout rate of 0.03 before finally outputting the predicted class. The performance of this model was tested by measuring the accuracy of its classifications compared to the plain-text results. The results were verified using tenfold cross validation with record randomization for each data set.

The ANN parameters were chosen based on empirical experimentation. While it is possible to achieve the same results with a smaller network, we decided to use a single model for all Hash-Comb configurations for two reasons. The first is to have a standardized design to test the performance and execution times of our scheme. The second is that it is undesirable from a practical standpoint to have a different model for each Hash-Comb resolution. The only difference between models, in this case, is the input layer size which has to be adjusted to be larger by ω.

A. RESULTS

The experimental results for the "Full" dataset are summarized in Figure 6 and 7. Starting with Group 1, we find that using all $\omega = 8$ dimensions h_1 through h_8 (Column: **All dimensions**) results in an accuracy of 97.81% (8 labels), 90.72% (16 labels) and 90.49% (24 labels). This decrease in accuracy with the increasing number of labels is due to these labels being generated using a clustering algorithm, having more clusters will naturally lead to closer borders. This increases the necessary QR needed to obtain high accuracy. Furthermore, classification problems with more labels are generally considered more difficult.

Removing the dimension in the last position (h_4 and h_8), which corresponds to the hash of the full IP address (Column: **Remove** $H_4 \mid H_8$) results in no noticeable change to the the accuracy (97.88%, 89.28% and 90.67%) which is expected since removing the last dimension is essentially removing a small piece of information which only helps in determining exact equality.

Removing two more dimensions (h_3 and h_7) starts to slightly degrade the performance, resulting in 96.91%, 84.77% and 86.79% accuracy (Columns: **Remove** $H_{3,4} \mid H_{7,8}$). Only at removing three dimensions does the performance drop (92.86%, 79.08% and 80.78%) (Columns: **Remove** $H_{2,3,4} \mid H_{6,7,8}$). An interesting result can be observed when removing all the comb's teeth except for the maximum QR dimensions (Columns: **Remove** $H_{1,2,3} \mid H_{5,6,7}$ where we see no noticeable degradation beyond the previous case (92.46%, 81.79% and 78.88%). Intuition tells us that hashing the full IP address should not be sufficient to obtain performance comparable to the previous cases. However, it is important to note that in the original dataset, the majority of network traffic tends to flow to a small number of IPs, which is causing this behaviour. Moving on to Group 2, we find that using $\omega = 8$ gives a noticeably improvement to accuracy (98.50% using 16 labels) compared to $\omega = 4$ (90.72%) which supports our intuition. We can observe this improvement across all the remaining tests in Group 2.

Since removing certain dimensions had no significant effect on accuracy, we can conclude that injecting random

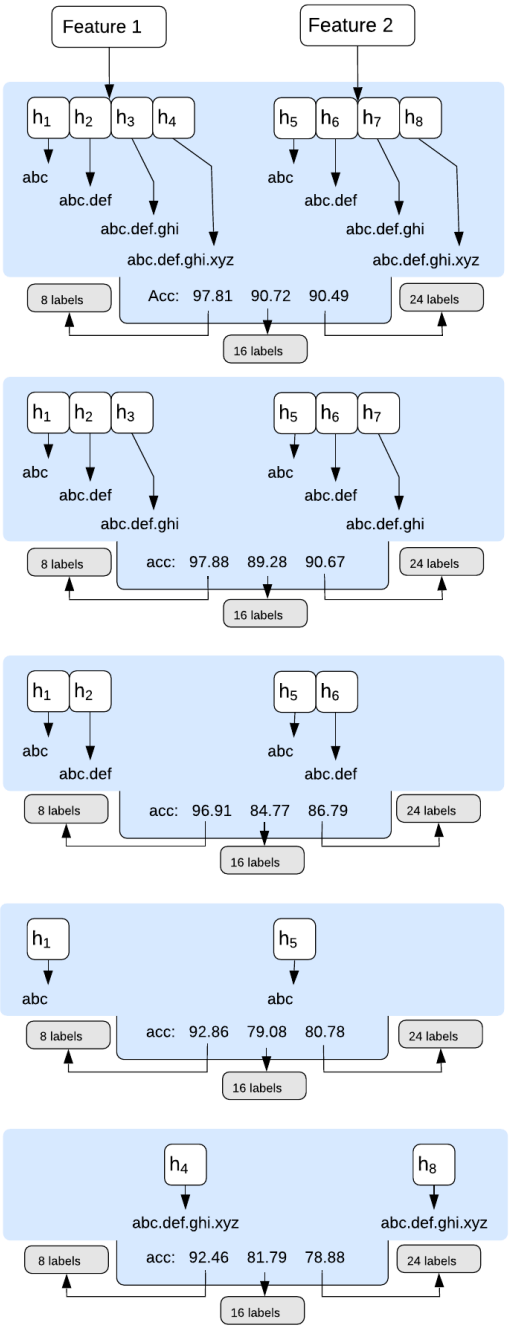


FIGURE 6. Summary of the results on the first version of the data set (part a).

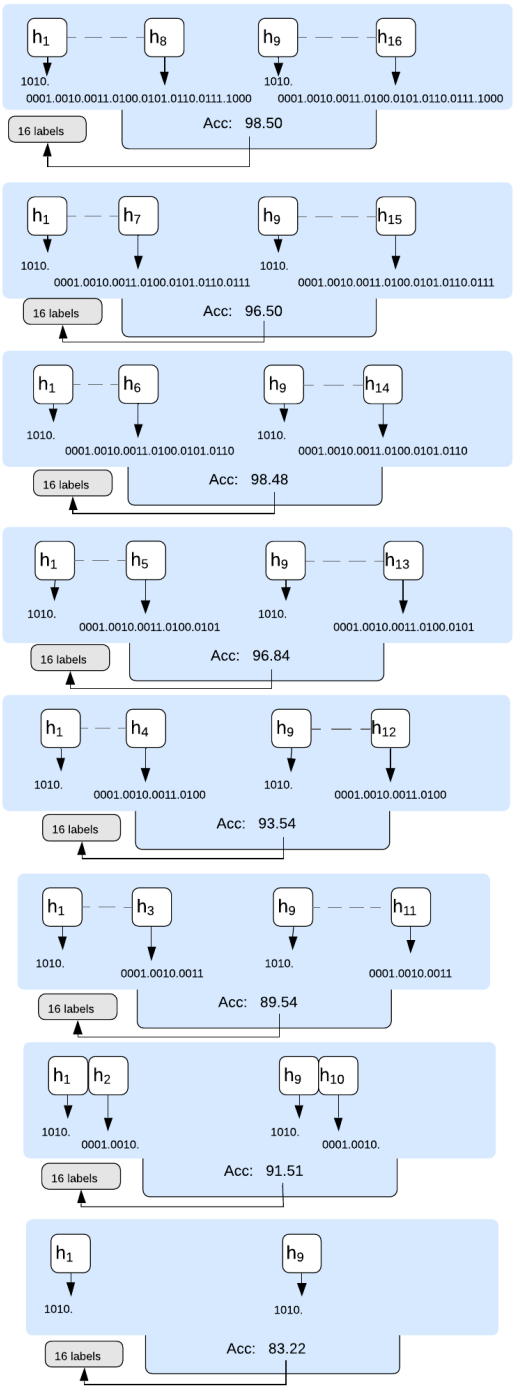


FIGURE 7. Summary of the results on the first version of the data set (part b).

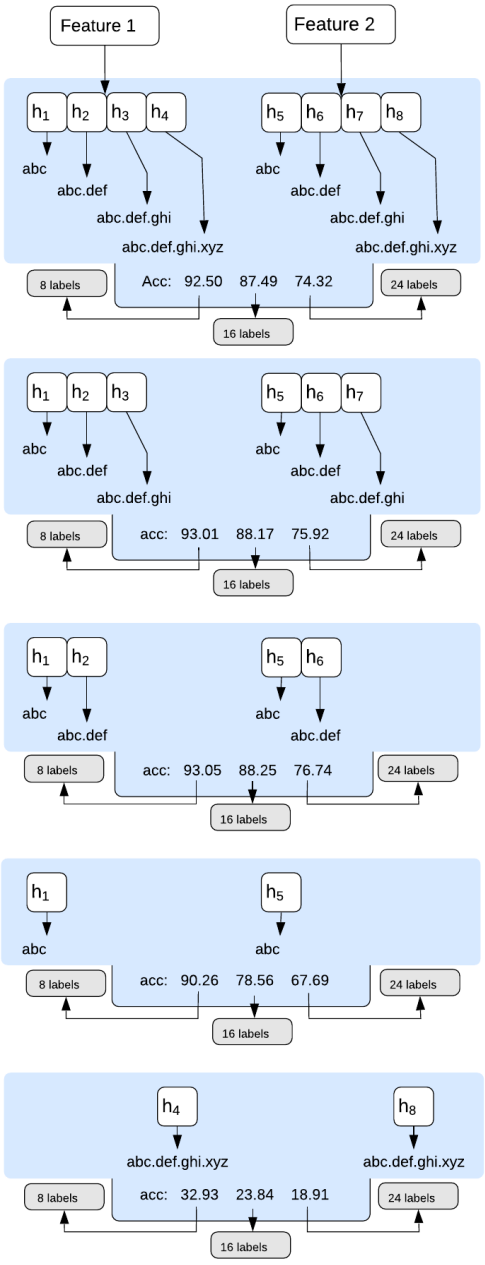


FIGURE 8. Summary of the results on the second version of the data set (part a).

noise into those lower tier dimensions will not cause any significant performance degradation.

The results for the second (''Diverse'') dataset, which only contains unique IP pairs is reported in Figure 8 and 9. As expected, the accuracy is lower due to the uniform distribution of unique IPs, which makes it harder for the quantization to group them. The highest accuracy achievable with $\omega = 4$ is 92.50% with 8 labels and 87.49% for 16 labels and 74.32% 24 labels. To obtain higher accuracy, we have to use $\omega = 8$ which produces an accuracy of 93.81% on 16 labels. Unlike the previous data set, this one highlights the advantage of using quantization as opposed to just hashing the data where traditional hashing gets an accuracy of 32.93%, 23.84% and 18.91%. The low accuracy is due to removing repeating IPs causing each hash in the data set to be unique, with no preserved distance between them.

Generally speaking, our experiment shows that the level of privacy of Hash-Combs depends on the interaction between the quantization process and the original - distribution [9].

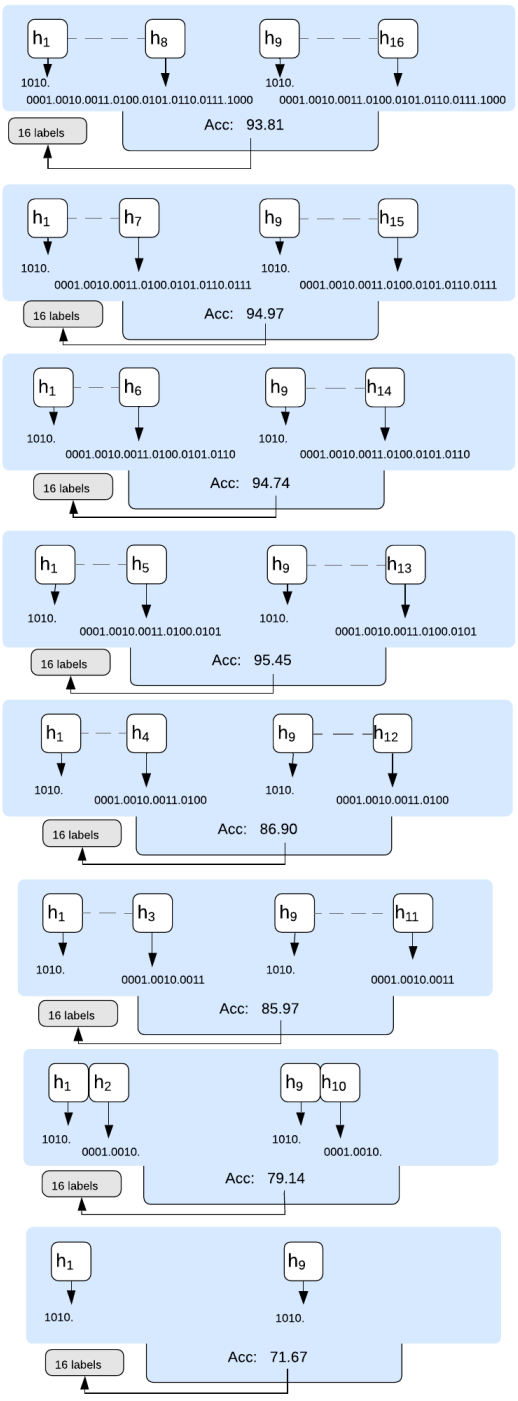


FIGURE 9. Summary of the results on the second version of the data set (part b).

Thus, it is possible to tweak the channel sizes to achieve a configuration that maximizes either accuracy or privacy but not both simultaneously.

V. CONCLUSION AND FUTURE WORK

In this paper, we addressed the challenge of performing data analytics on hashed data while complying with data privacy regulations. We presented Hash-Combs, a hierarchical distance-preserving hash scheme that can achieve regulation-compliance by using certified versions of basic hash libraries. The paper described in detail the generation of our multi-hash data representation and demonstrated how it can be used to enable distance-based analytics. Experimentation involved a real data set towards detecting online advertisement fraud and 84 test scenarios. Hash-Combs returned favourable results of up to 97.88% accuracy compared to the plain-text analytics under selected configurations.

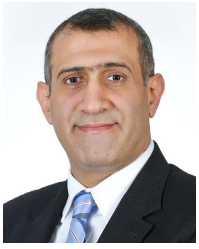
REFERENCES

[1] A. Bilal, S. Wingreen, and R. Sharma, "Virtue ethics as a solution to the privacy paradox and trust in emerging technologies," in *Proc. 3rd Int. Conf. Inf. Sci. Syst.*, Mar. 2020, pp. 224–228.
[2] Commission Nationale de l'Informatique et des Libertes (CNIL), *Security of Personal Data*, The Cnil's Guides, Paris, France, 2018.
[3] A. Jelinek, "Statement of the WP29 on encryption and their impact on the protection of individuals with regard to the processing of their personal data in the EU," Eur. Data Protetion Board, Brussels, Belgium, Tech. Rep. 4/2018, Apr. 2018.
[4] UK's Information Commision Office (ICO), *Encryption*, Information Commissioner's Office, Wilmslow, U.K., Jan. 2019.
[5] G. Stephen, "Encryption may lower fines under new EU privacy regime," *Bloomberg Law*, to be published. [Online]. Available: https://news.bloombergtax.com/privacy-and-data-security/encryption-may-lower-fines-under-new-eu-privacy-regime?context=search&index=9
[6] W. Knight, "The dark secret at the heart of AI," MIT Technol. Rev., Cambridge, MA, USA, Tech. Rep., 2017.
[7] X. Sun, P. Zhang, J. K. Liu, J. Yu, and W. Xie, "Private machine learning classification based on fully homomorphic encryption," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 2, pp. 352–364, Jun. 2020.
[8] S. Cimato and E. Damiani, "Some ideas on privacy-aware data analytics in the internet-of-everything," in *From Database to Cyber Security*. Cham, Switzerland: Springer, 2018, pp. 113–124.
[9] S. Salamatian, A. Zhang, F. d P. Calmon, S. Bhamidipati, N. Fawaz, B. Kveton, P. Oliveira, and N. Taft, "How to hide the elephant- or the donkey- in the room: Practical privacy against statistical inference for large data," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Austin, TX, USA, Dec. 2013, pp. 269–272.
[10] M. C. Compagnucci, J. Meszaros, T. Minssen, A. Arasilango, T. Ous, and M. Rajarajan, "Homomorphic encryption: The 'Holy grail' for big data analytics and legal compliance in the pharmaceutical and healthcare sector?" *EPLR*, vol. 3, p. 144, Mar. 2019.
[11] R. Badhwar, "The need for post-quantum cryptography," in *The CISO's Next Frontier*. Cham, Switzerland: Springer, 2021, pp. 15–30.
[12] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. Brendan McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for federated learning on user-held data," 2016, *arXiv:1611.04482*.
[13] M. Cunha, R. Mendes, and J. P. Vilela, "A survey of privacy-preserving mechanisms for heterogeneous data types," *Comput. Sci. Rev.*, vol. 41, Aug. 2021, Art. no. 100403.
[14] A. Hussain, L. A. Lasrado, R. R. Mukkamala, and U. Tanveer, "Sharing is caring–design and demonstration of a data privacy tool for interorganizational transfer of data," *Proc. Comput. Sci.*, vol. 181, pp. 394–402, Jan. 2021.
[15] M. Cappellari, J. Belstner, B. Rodriguez, and J. Sedayao, "A cloud-based data collaborative to combat the COVID-19 pandemic and to solve major technology challenges," *Future Internet*, vol. 13, no. 3, p. 61, Feb. 2021.
[16] Y. Yang, F. Chen, X. Zhang, J. Yu, and P. Zhang, "Research on the hash function structures and its application," *Wireless Pers. Commun.*, vol. 94, no. 4, pp. 2969–2985, Jun. 2017.
[17] M. Martín-Pérez, R. J. Rodríguez, and F. Breitinger, "Bringing order to approximate matching: Classification and attacks on similarity digest algorithms," *Forensic Sci. Int., Digit. Invest.*, vol. 36, Apr. 2021, Art. no. 301120.
[18] J. Kornblum, "Identifying almost identical files using context triggered piecewise hashing," *Digit. Invest.*, vol. 3, pp. 91–97, Sep. 2006.
[19] C. Zheng, X. Li, Q. Liu, Y. Sun, and B. Fang, "Hashing incomplete and unordered network streams," in *Advances in Digital Forensics XIV*, G. Peterson and S. Shenoi, Eds. Cham, Switzerland: Springer, 2018, pp. 199–224.
[20] V. Roussev, G. G. Richard, and L. Marziale, "Multi-resolution similarity hashing," *Digit. Invest.*, vol. 4, pp. 105–113, Sep. 2007.

[21] F. Breitinger and H. Baier, "Similarity preserving hashing: Eligible properties and a new algorithm MRSH-v2," in *Digital Forensics and Cyber Crime*, M. Rogers and K. C. Seigfried-Spellar, Eds. Berlin, Germany: Springer, 2013, pp. 167–182.
[22] D. Lillis, F. Breitinger, and M. Scanlon, "Expediting MRSH-v2 approximate matching with hierarchical Bloom filter trees," in *Digital Forensics and Cyber Crime*, vol. 216, P. Matousek and M. Schmiedecker, Cham, Eds. Switzerland: Springer, 2018, pp. 144–157.
[23] F. Breitinger and H. Baier, "A fuzzy hashing approach based on random sequences and Hamming distance," in *Proc. Conf. Digit. Forensics, Secur. Law*, 2012, pp. 89–100.
[24] D. Chang, M. Ghosh, S. K. Sanadhya, M. Singh, and D. R. White, "FbHash: A new similarity hashing scheme for digital forensics," *Digit. Invest.*, vol. 29, pp. S113–S123, Jul. 2019.
[25] E. Damiani, S. D. C. D. Vimercati, S. Paraboschi, and P. Samarati, "An open digest-based technique for spam detection," *ISCA PDCS*, 2004, pp. 559–564.
[26] B. Charyyev and M. H. Gunes, "Detecting anomalous IoT traffic flow with locality sensitive hashes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.
[27] J. Oliver, C. Cheng, and Y. Chen, "TLSH–A locality sensitive hash," in *Proc. 4th Cybercrime Trustworthy Comput. Workshop*, Nov. 2013, pp. 7–13.
[28] M. Ali, J. Hagen, and J. Oliver, "Scalable malware clustering using multistage tree parallelization," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Nov. 2020, pp. 1–6.
[29] F. Breitinger, K. P. Astebol, H. Baier, and C. Busch, "MvHash-B–A new approach for similarity preserving hashing," in *Proc. 7th Int. Conf. IT Secur. Incident Manage. IT Forensics*, Mar. 2013, pp. 33–44.
[30] A. AlMahmoud, M. Colombo, C. Y. Yeun, and H. Al-Muhairi, "Enhancement of key derivation in web service security," *Wireless Pers. Commun.*, vol. 97, no. 4, pp. 5171–5184, Dec. 2017.
[31] C. Winter, M. Schneider, and Y. Yannikos, "F2S2: Fast forensic similarity search through indexing piecewise hash signatures," *Digit. Invest.*, vol. 10, no. 4, pp. 361–371, Dec. 2013.
[32] W. Kong, W.-J. Li, and M. Guo, "Manhattan hashing for large-scale image retrieval," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. - SIGIR*, Portland, OR, USA, 2012, p. 45.
[33] Z. Wang, L.-Y. Duan, J. Lin, X. Wang, T. Huang, and W. Gao, "Hamming compatible quantization for hashing," in *Proc. 24th Int. Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 2298–2304.
[34] W. Liu, C. Mu, S. Kumar, and S.-F. Chang, "Discrete graph hashing," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Montreal, QC, Canada: Curran Associates, 2014, pp. 3419–3427.
[35] J. Xu, P. Wang, G. Tian, B. Xu, J. Zhao, F. Wang, and H. Hao, "Convolutional neural networks for text hashing," in *Proc. 24th Int. Conf. Artificial Intell.*, Buenos Aires, Argentina, 2015, pp. 1369–1375.
[36] R. Schwartmann, "Guidelines for the legally secure deployment of pseudonymization solutions in compliance with the general data protection regulation, version 1.0," Data Protection Focus Group, German Assoc. Data Protection Data Secur., Bonn, Germany, Tech. Rep., 2017.
[37] M. Holusa and E. Sojka, "A K-max geodesic distance and its application in image segmentation," in *Computer Analysis of Images and Patterns* (Lecture Notes in Computer Science), G. Azzopardi and N. Petkov, Eds. Cham, Switzerland: Springer, 2015, pp. 618–629.
[38] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 70–91, 1st Quart., 2015.
[39] K. Crane, C. Weischedel, and M. Wardetzky, "Geodesics in heat: A new approach to computing distance based on heat flow," *ACM Trans. Graph.*, vol. 32, no. 5, p. 152, Oct. 2013.
[40] S. Gattone, A. De Sanctis, S. Puechmorel, and F. Nicol, "On the geodesic distance in shapes K-means clustering," *Entropy*, vol. 20, no. 9, p. 647, Aug. 2018.
[41] A. Criminisi and J. D. J. Shotton, "Semi-supervised random decision forests for machine learning using Mahalanobis distance to identify geodesic paths," U.S. Patent 9 519 868 B2, Dec. 13, 2016.
[42] O. Rottenstreich and J. Tapolcai, "Optimal rule caching and lossy compression for longest prefix matching," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 864–878, Apr. 2017.
[43] A. AlMahmoud, E. Damiani, H. Otrok, and Y. Al-Hammadi, "Spamdoop: A privacy-preserving big data platform for collaborative spam detection," *IEEE Trans. Big Data*, vol. 5, no. 3, pp. 293–304, Sep. 2019.
[44] *The Bot Baseline: Fraud in Digital Advertising 2017 Report | ANA*, Assoc. Nat. Advertisers, New York, NY, USA, 2017.
[45] M. Abadi, P. Barham, J. Chen, Z. Chen, and A. Davis, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

ABDELRAHMAN ALMAHMOUD (Member, IEEE) is currently the Principal Cloud and Big Data Researcher at the Technology Innovation Institute (TII). He is also leading the Secure Cloud and Machine Learning Group, TII's Secure Systems Research Centre [SSRC]. He worked in the research domain for close to ten years covering topics, such as big data analytics paradigms, privacy-preserving analytics/machine learning and cyber security. He worked with the Artificial Intelligence Office in the Prime Minister's Office as an Advisor/Head of Research and Infrastructure where he worked on several national and international initiatives. He was also a member of the first UAE Youth Science Council, a Scientific Sub-Committee Member of the Sheikh Hamdan Bin Rashid Al Maktoum Award for Medical Sciences (11th term, AI in Medicine), and a member of the UAE's Steering Committee for the ISO SC42 working group for AI. He is a Technical Committee Member of The Digital School initiative under Mohammed Bin Rashid Al Maktoum Global Initiatives.

ERNESTO DAMIANI (Senior Member, IEEE) is currently a Full Professor at the Università degli Studi di Milano, Italy, a Senior Director of the Robotics and Intelligent Systems Institute, and the Director of the Center for Cyber Physical Systems (C2PS), Khalifa University, United Arab Emirates. He is the Leader of the Big Data Area, Etisalat British Telecom Innovation Center (EBTIC), and the President of the Consortium of Italian Computer Science Universities (CINI). He is also part of the ENISA Ad-Hoc Working Group on Artificial Intelligence Cybersecurity. He has pioneered model-driven data analytics. He has authored more than 650 Scopus-indexed publications and several patents. His research interests include cyber-physical systems, big data analytics, edge/cloud security and performance, artificial intelligence, and machine learning. He was a recipient of the Research and Innovation Award from the IEEE Technical Committee on Homeland Security, of the Stephen Yau Award from the Service Society, of the Outstanding Contributions Award from IFIP TC2, of the Chester-Sall Award from IEEE IES, of the IEEE TCHS Research and Innovation Award, and of a Doctorate Honoris Causa from INSA-Lyon, France, for his contribution to big data teaching and research.



HADI OTROK (Senior Member, IEEE) received the Ph.D. degree in ECE from Concordia University. He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science (EECS), Khalifa University. He is also an Affiliate Associate Professor with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada, and an Affiliate Associate Professor with the Electrical Department, Ecole de Technologie Superieure (ETS), Montreal. His research interests include the domain of blockchain, reinforcement learning, crowd sensing and sourcing, ad hoc networks, and cloud security. He co-chaired several committees at various IEEE conferences. He is also an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, *Ad Hoc Networks* (Elsevier), and *IEEE Network*. From 2015 to 2019, he also served as an Associate Editor for the IEEE COMMUNICATIONS LETTERS.