On privacy preserving search in large scale distributed systems: a signal processing view on searchable encryption

Sviatoslav Voloshynovskiy*, Fokko Beekhof, Oleksiy Koval, and Taras Holotyak

University of Geneva, Department of Computer Science
7 route de Drize, CH 1227, Geneva, Switzerland
{svolos,fokko.beekhof,oleksiy.koval,taras.holotyak}@unige.ch
http://sip.unige.ch

Abstract. In this paper, we advocate an alternative signal processing based approach to searchable encryption architectures allowing to find non-exact or similar matches in the encrypted domain. The proposed approach is based on a modified architecture, where the main computational load is reallocated to a data user, who challenges an unsecure server by multiple requests, while the role of the server is reduced to appropriately replying to these challenges. To minimize the number of challenges per query, we propose a concept of bit reliability allowing to filter out the most reliable bits to formulate the most precise query in the shortest number of steps that can match the encrypted counterpart stored in the server database. Several practical implementations are discussed and empirical upper bounds on the search accuracy in terms of average probability of error are obtained for real image search under various distortions including additive Gaussian noise, uniform noise and lossy JPEG compression.

Key words: privacy preserving search, searchable encryption, similar matching, dimensionality reduction, bit reliability.

1 Introduction

Modern information management systems are characterized by the highly distributed character of information acquisition, processing, storage and access where certain parts of these systems are outsourced to service-specialized third parties that are not always trustworthy. The examples of these architectures are numerous and include outsourcing of email services, multimedia data, electronic libraries, P2P data sharing systems, stock exchanges, medicine, bioinformatics, identification of people based on biometrics and physical objects based on physical unclonable functions, etc. In many of these applications, the data owner stores some possibly privacy sensitive information on an untrusted server, which models the outsourced service provider. At the same time, this data should be

* The contact author is S. Voloshynovskiy.

provided to several authorized users, who are allowed to access and search it. The data owner and data user might possess a common secret. Since the server might be honest-but-curious or even malicious, the data provided by the data owner to the server may be encrypted using the above common secret. This part addresses the security-privacy issue of the stored data that should not reveal any security-privacy leaks to the server or any unauthorized users.

At the same time, the data user query for the information search should reveal as little information as possible to the server and unauthorized parties about both the stored data itself and query of interest to the data user. Moreover, the data user might posses only some inexact, distorted or noisy equivalent of the data owner's data. Thus, exact matching strategies are not always applicable. Additionally, the number of communication sessions per query might be constrained due to communication, security or privacy reasons. The basic model of such a distributed search system is shown in Fig. 1. The main challenge behind

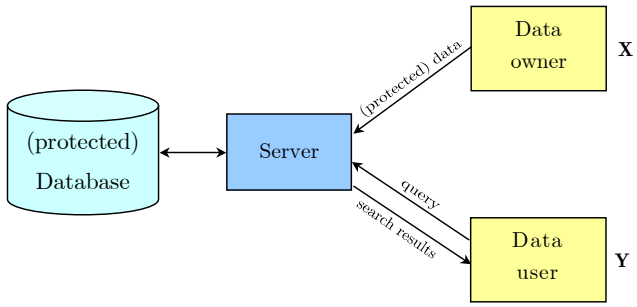

Fig. 1. Basic block-diagram of privacy preserving information search system.

this system consists in the privacy preserving information search in the space of protected data. In the case that all information is stored in the encrypted form in the database, one possible approach consists in downloading the entire database, decrypt it, and performing the data search locally on the plaintext data. This is usually unacceptable, since the database might be too large, quickly obsolete, confidential or privacy sensitive.

Another solution known as a *searchable encryption* consists in searching directly in the protected data at the server. Assuming that the entire database is stored in the encrypted form, the data user can encrypt his query and send it to the server. If the server finds a match, the corresponding encrypted database entry is sent to the data user, who can decrypt it. Not pretending to be exhaustive in our review, we refer the reader to [1] for the classification of different secure search strategies in the encrypted domain. The existing techniques can be classified in four groups: *indices* based search, when the actual search is performed on an added index (hash of encrypted data) [2–4]; *trapdoor encryption* based search, when the search is preformed based on a predefined codebook of encrypted code-

words [5–7]; *secret sharing* based search, when the data is distributed over several servers, which are assumed to not collude [8]; and *homomorphic encryption* based search, when the search is performed directly in the encrypted domain using similarity functions in the class of homomorphic encryption [9–11]. Being attractive in terms of both communication and security, all these existing techniques currently work only for exact matches. However, in many applications the query might be not well defined and only an inexact copy of the sought database entry may be given due to the lack of reliable priors, ambiguity about the agreed codebooks, keywords or templates, fast and frequent database updates with the appearance of new entries, noise or distortions during the acquisition or lossy compression during storage or communication. Therefore, the data user might be interested in finding some similar entries in the database. Additionally, practically all the above search strategies are linear with respect to the database size M, i.e., $O(M)$, that might be prohibitively large in certain applications.

That is why the goal of this paper is to consider a privacy preserving protocol enabling search in the protected domain based on possibly distorted or inexact queries with a limited number of communication sessions per query. We will essentially refer to the signal processing and communication formulation of this problem. The data user, as the final receiver of the information, should be able to conceal his query and to obtain either a single best match or a list of the most similar matches in the protected database. We will exemplify the consideration referring to the image search in the encrypted domain. Along this consideration, we will assume that the image represents some sort of privacy sensitive part of a plaintext database (fingerprint, iris, photo, medical examination image, etc.,) while the associated identification information (ownership, date, place of creation, results of examination, etc.) constitutes to the secret part. We will also assume that the identification information can be disclosed to the data user, while the privacy sensitive part should not be disclosed to anyone.

Our setting is close to the private search based on searchable encryption where the role of data owner, server, data user and unauthorized users are considered above. A nice summary on the objectives of these players can be found in the SPEED project report [12]. In application to our problem formulation, these objectives can be restated as follows:

- **Data owner**: The data owner provides the plaintext database containing private information, which should be accessible only by the authorized users. The data owner encrypts the plaintext database in part of identification data and computes the hashes from privacy sensitive parts according to the corresponding key management protocol. The resulting protected database is stored at a possibly untrusted server. The data owner is interested in the correct retrieval of identification information by the data user based on his query but not in the disclosure or distribution of the private part of database.
- **Server**: The server stores the protected database, processes and correctly answers the queries of data users. The server does not know any secret key. The server may be able to track the users' access patterns.

- **Data users**: Data users possess the secret key used by the data owner that enables them to generate valid queries and to decrypt the identification part of database entries found by the server on their request.
- **Unauthorized users**: Unauthorized users do not know the protected database or any currently used secret. An unauthorized user can observe the queries from the authorized users and corresponding server answers that might help him learn (at least partially) the protected database.

The *security-privacy requirements* follow from the players' objectives:

- **Data user**: The identification the part of the plaintext database can be shared with the data users who submit valid queries. The private part stored in the database should not leak any information even to the data users who do not possess the valid query.
- **Server**: An honest-but-curious or even malicious server should not be able to gain any information on the plaintext of the encrypted database based on both stored data and the data users' queries.
- **Unauthorized users**: Unauthorized users should not gain any information on the plaintext database from any strategy: (a) direct server querying and analyzing the server responses, (b) colluding with the server and (c) analyzing the authorized data user's queries and the corresponding responses.

In addition, the *robustness requirement* assumes that the data users should be able to retrieve the best match with the protected privacy sensitive plaintext data in some metric space with the defined matching score.

The main contribution of this paper consists in the symmetric protocol for privacy preserving search based on non-exact or fuzzy matching and a concept of bit reliability. The concept of bit reliability proposed in the context of private information search makes possible to reduce the complexity and the number of communication sessions per query. Moreover, it naturally reflects the fact of informed collaboration between the data owner and data user.

The paper has the following structure. Section 2 reviews the optimal match strategies for known and unknown models of data and query statistical dependence as well as presents the results for the performance in the feature space of reduced dimensionality as well as introduces a generic search strategy in the encrypted domain. The proposed approach is presented in Section 3. Section 4 introduces a concept of bit reliability for the enhanced search strategies and the experimental results are presented in Section 6. Finally, Section 7 concludes the paper.

Notations: We use capital letters to denote scalar random variables X and $\mathbf{X}$ to denote vector random variables, corresponding small letters x and $\mathbf{x}$ to denote the realizations of scalar and vector random variables, respectively. All vectors without sign tilde are assumed to be of the length N and with the sign tilde of length L with the corresponding subindexes. The binary representation of vectors will be denoted as $b_{\mathbf{x}}$ with the corresponding subindexing. We use $\mathbf{X} \sim p_{\mathbf{X}}(\mathbf{x})$ or simply $\mathbf{X} \sim p(\mathbf{x})$ to indicate that a random variable $\mathbf{X}$ is distributed according to $p_{\mathbf{X}}(\mathbf{x})$. $\mathcal{N}(\mu, \sigma_X^2)$ stands for Gaussian distribution with mean μ and variance σ_X^2. $||.||$ denotes Euclidean vector norm and $Q(.)$ stands for Q-function.

2 Search strategies

2.1 Optimal maximum likelihood search

We will assume that the data owner has M entries in the database indexed by an index m, i.e., $\mathbf{x}(m) \in \mathbb{R}^N$, $1 \leq m \leq M$. The index m is associated to all identification information discussed in the introduction and the data $\mathbf{x}(m)$ is some privacy sensitive part of the database.

At the same time, the data user has some query data $\mathbf{y} \in \mathbb{R}^N$ that can be in relationship with some $\mathbf{x}(m)$ via a probabilistic mapping $p(\mathbf{y}|\mathbf{x})$. The data user wishes to retrieve the identification information of $\mathbf{x}(m)$ that is the closest to the query $\mathbf{y}$. In the non-secure setting, this can be resolved by the maximum likelihood (ML) rule:

$$\hat{m} = \arg \max_{1 \leq m \leq M} p(\mathbf{y}|\mathbf{x}(m)), \tag{1}$$

assuming that all entries are equilikely and the probabilistic relationship $p(\mathbf{y}|\mathbf{x})$ is known. Thus, the server can simply check out the entire database in $O(M)$ trials and find a single m that satisfies (1). To prevent matching with the queries not contained in the original database, one can additionally impose the constraint on consistency:

$$\hat{m} = \arg \max_{1 \leq m \leq M : p(\mathbf{y}|\mathbf{x}(m)) \geq r_p} p(\mathbf{y}|\mathbf{x}(m)), \tag{2}$$

where r_p is the threshold that ensures that the result is found among the correct entries.

The statistical relationship can be naturally converted into the metric space as $d(\mathbf{y}, \mathbf{x}(m)) = -\ln p(\mathbf{y}|\mathbf{x}(m))$ that converts (2) into:

$$\hat{m} = \arg \min_{1 \leq m \leq M : d(\mathbf{y}, \mathbf{x}(m)) \leq r_d} d(\mathbf{y}, \mathbf{x}(m)), \tag{3}$$

where $r_d \equiv -\ln r_p$ corresponds to matching radius. In the case of multiple matching, a so-called list decoding, the minimization can be abandoned and only the condition $d(\mathbf{y}, \mathbf{x}(m)) \leq r_d$ should be satisfied.

Depending on the particular $p(\mathbf{y}|\mathbf{x})$ one can deduce the ℓ_2 (Euclidian) metric for the Gaussian conditional probability density function (pdf), the ℓ_1 (absolute) metric for Laplacian pdfs or, more generally, the ℓ_p (Sobolev) metric for Generalized Gaussian pdfs.

The search is normally performed in the defined feature space of lower dimensionality $\tilde{\mathbf{x}}(m) \in \mathbb{R}^L$ and $\tilde{\mathbf{y}} \in \mathbb{R}^L$ obtained from $\mathbf{x}(m)$ and $\mathbf{y}$ by some dimensionality reduction transform:

$$\tilde{\mathbf{x}}(m) = \mathbf{W}\mathbf{x}(m), \tag{4}$$

$$\tilde{\mathbf{y}} = \mathbf{W}\mathbf{y}, \tag{5}$$

where $\mathbf{W} \in \mathbb{R}^{L \times N}$ and $L \leq N$ and $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_L)^T$ consists of a set of projection basis vectors $\mathbf{w}_i \in \mathbb{R}^N$ with $1 \leq i \leq L$. In this case, (3) is reduced to the search in the space of feature vectors:

$$\hat{m} = \arg \min_{1 \leq m \leq M : \tilde{d}(\tilde{\mathbf{y}}, \tilde{\mathbf{x}}(m)) \leq \tilde{r}_d} \tilde{d}(\tilde{\mathbf{y}}, \tilde{\mathbf{x}}(m)), \tag{6}$$

where $\tilde{d}(.,.)$ denotes the corresponding metric in the feature space. The feature space can also be design to ensure the invariance to certain geometrical desynchronizations.

2.2 Universal search and its approximations

When the pdf $p(\mathbf{y}|\mathbf{x}(m))$ is unknown, the ML search can not be used. In fact, one can demonstrate that the search results critically depends on the selection of the metric. For example, the popular nearest-neighborhood strategy or maximum correlation coefficient search can provide extremely poor performance if the model $p(\mathbf{y}|\mathbf{x}(m))$ deviates from the Gaussian pdf. One can refer to the case of image search in the database of compressed or blurred images, where the above search strategies will be sub-optimal. The problem consists in the fact that the lossy compression and blurring distortions belong to the group of non-Gaussian distortions and are even image dependent in the case of lossy image compression.

A particular solution to this problem consists in the usage of maximum mutual information (MMI) search, which provides universally attainable error exponents for random-composition databases over discrete memoryless mappings, i.e., $p(\mathbf{y}|\mathbf{x}(m)) = \prod_{i=1}^{N} p(y_i|x_i(m))$. Since the random-composition condition on the database entries can be hardly verified in practice and the mapping $p(\mathbf{y}|\mathbf{x}(m))$ is not always memoryless (like compression or blurring examples), we will follow another strategy in this paper.

The main idea behind the proposed approach consists in the transformation of the database entries into the domain with the predefined properties. This sequentially leads to the transformation of the corresponding metric from the direct observation space into the predefined metric in the transformation space. Sequentially, one can perform the search according to the optimal known metric. Moreover, this transformation can be naturally combined with the feature extraction step (4).

In the scope of this paper, we will assume that the transformation is performed based on any randomized orthogonal matrix $\mathbf{W}$ whose elements $w_{i,j}$ are generated from some specified distribution. An $L \times N$ random matrix $\mathbf{W}$ whose entries $w_{i,j}$ are independent realizations of Gaussian random variables $W_{i,j} \sim \mathcal{N}(0, \frac{1}{N})$ presents a particular interest for our study. In this case, such a matrix can be considered as an almost *orthoprojector*, for which $\mathbf{W}\mathbf{W}^T \approx \mathbf{I}_L$ [1]. The selection of basis vectors with Gaussian distribution also guarantees the Gaussian distribution of projected coefficients under sufficiently large N according to the Central Limit Theorem. This means that the search rule (6) in the generic projected domain can be reduced to the minimum distance rule:

$$\hat{m} = \arg \min_{1 \le m \le M : s.t. \|\tilde{\mathbf{y}} - \tilde{\mathbf{x}}(m)\|^2 \le \tilde{r}_d} \|\tilde{\mathbf{y}} - \tilde{\mathbf{x}}(m)\|^2 . \quad (7)$$

Furthermore, to enable the use of cryptographic primitives and obtain a fast search procedure it is also beneficial to perform the search in the binary space

[1] Otherwise, one can apply special orthogonalization techniques to ensure perfect orthogonality.

that can be achieved by a *binarization*. The most simple binarization of extracted features can be performed as:

$$b_{\mathbf{x}_i} = sign(\mathbf{w}_i^T \mathbf{x}), \tag{8}$$

where $b_{\mathbf{x}_i} \in \{0,1\}$, with $1 \le i \le L$ and $sign(a) = 1$, if $a \ge 0$ and 0, otherwise. The vector $\mathbf{b_x} \in \{0,1\}^L$ computed for all projections represents a *binary template* of the vector $\mathbf{x}$. Since all projections are independent, it can be assumed that all bits in $\mathbf{b_x}$ will be independent and equiprobable.[2] In this case, the search strategy is reduced to:

$$\hat{m} = \arg \min_{1 \le m \le M : d^H(\mathbf{b_y}, \mathbf{b_x}(m)) \le \tilde{r}_b} d^H(\mathbf{b_y}, \mathbf{b_x}(m)), \tag{9}$$

where $d^H(.,.)$ denotes the Hamming distance and $\tilde{r}_b$ corresponds to the matching radius in the binary space.

The template computed from some distorted version $\mathbf{y}$ of $\mathbf{x}$ denoted as $\mathbf{b_y}$ might contain some bits different from those in $\mathbf{b_x}$. Therefore, the link between the binary representation $\mathbf{b_x}$ of vector $\mathbf{x}$ and its noisy counterpart $\mathbf{b_y}$ of vector $\mathbf{y}$ is defined according to the binary symmetric channel (BSC) with average bit error probability $\bar{P}_b$. The bit error probability indicates the mismatch of signs between $\tilde{x}_i$ and $\tilde{y}_i$ according to (8), i.e., $\Pr[sign(\tilde{x}_i) \neq sign(\tilde{y}_i)]$. For a given $\mathbf{x}$ and $\mathbf{w}_i$, the probability of bit error is:

$$P_{b|\tilde{x}_i} = \frac{1}{2}(\Pr[\tilde{Y}_i \ge 0 | \tilde{X}_i < 0] + \Pr[\tilde{Y}_i < 0 | \tilde{X}_i \ge 0]), \tag{10}$$

or by symmetry as:

$$P_{b|\tilde{x}_i} = \Pr[\tilde{Y}_i < 0 | \tilde{X}_i \ge 0]. \tag{11}$$

We will exemplify the consideration for the i.i.d. Gaussian setup with $\mathbf{y} = \mathbf{x} + \mathbf{z}$ and $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \sigma_X^2 \mathbf{I}_N)$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \sigma_Z^2 \mathbf{I}_N)$. For a given $\tilde{x}_i$ and Gaussian noise [3], the distribution of the projected vector is $\tilde{Y}_i \sim \mathcal{N}(\tilde{x}_i, \sigma_Z^2 \mathbf{w}_i^T \mathbf{w}_i)$ that reduces to $\tilde{Y}_i \sim \mathcal{N}(\tilde{x}_i, \sigma_Z^2)$ for the orthoprojector ($\mathbf{w}_i^T \mathbf{w}_i = 1$) and:

$$P_{b|\tilde{x}_i} = \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi\sigma_Z^2}} e^{\frac{-(\tilde{y}_i - \tilde{x}_i)^2}{2\sigma_Z^2}} d\tilde{y}_i = Q\left(\frac{\tilde{x}_i}{\sigma_Z}\right). \tag{12}$$

The above analysis only refers to a single realization of $\mathbf{x}$. Since $\mathbf{X}$ is a random vector following some distribution $p(\mathbf{x})$, one should find the average probability of error for all possible realizations. Assuming $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \sigma_X^2 \mathbf{I}_N)$, the statistics

[2] This assumption is only possible for independent input data. Since the transformed vectors will follow the Gaussian pdf but will not necessarily be decorrelated, one can apply the principle component analysis to decorrelate them, that, for the case of Gaussian data, will also provide their independence.

[3] In the case of assumed Gaussian random basis vectors $\mathbf{w}_i$ any distribution will be mapped into Gaussian one for both entry and noisy data.

of data in the projection domain are $\tilde{X}_i \sim \mathcal{N}(0, \sigma_X^2)$ and:

$$\bar{P}_b = 2\int_0^\infty P_{b|\tilde{x}_i} p(\tilde{x}_i) d\tilde{x}_i \tag{13}$$

$$= 2\int_0^\infty Q\left(\frac{\tilde{x}_i}{\sigma_Z}\right) \frac{1}{\sqrt{2\pi\sigma_X^2}} e^{\frac{-\tilde{x}_i^2}{2\sigma_X^2}} d\tilde{x}_i = \frac{1}{\pi}\arccos(\rho_{XY}), \tag{14}$$

where $\rho_{XY}^2 = \frac{\sigma_X^2}{\sigma_X^2+\sigma_Z^2}$ is the squared correlation coefficient between X and Y.

It should be also pointed out that the described procedure of binary template extraction represents a sort of privacy protection technique known as *sketches* [13]. Sketch calculation combines multiplicative randomization, i.e., dimensionality reduction with key-defined projection matrix, with the following binarization.

Obviously, the dimensionality reduction and binarization will effect the accuracy of search problem and more details can be found in [14]. However, the loss in performance can be greatly compensated by resolving the prior ambiguity about exact data models, security, privacy and search complexity benefits.

2.3 Generic search in the encrypted domain

According to the considered setup, the entire database is partitioned into two parts, i.e., the identification part denoted as $\mathrm{ID}(m)$ and a privacy sensitive part denoted as $\mathbf{x}(m)$, which is converted into some real feature space $\tilde{\mathbf{x}}(m)$ or binary space $\mathbf{b}_\mathbf{x}(m)$. A diagram of the search in the binary domain is shown in Fig. 2. Given the transform domain query $\mathbf{b}_\mathbf{y}$ and a specified search rule (9),

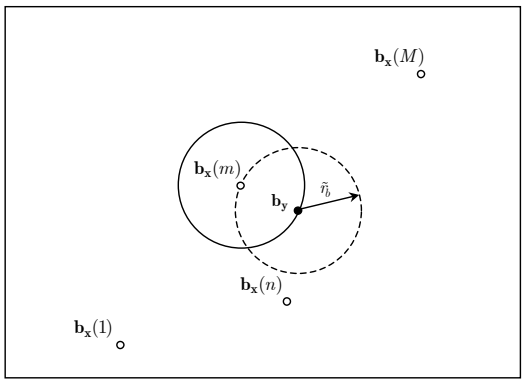

Fig. 2. Basic diagram of search in feature space: the algorithm should compute the distances to all codewords and select the closest within the defined radius.

one can retrieve either the closest vector $\mathbf{b}_\mathbf{x}(m)$ or all codewords in the specified neighborhood space. Such kind of search system faces several problems. First,

the complexity of the search is $O(M)$ that could be a serious restriction for large databases. Second, to ensure the security requirement the database should be encrypted. For this purpose, one can use a semantically secure homomorphic additive encryption approach, which allows multiplications of encrypted values by known variables from the encrypted query. However, in this case the complexity remains the same. Third, once a match or possibly plural matches are found the encrypted privacy sensitive part is sent to the data user for further decryption. However, the decryption of the privacy sensitive part is undesirable for many discussed reasons. One possible solution to this problem consists in the computation of a cryptographic hash from the privacy sensitive part. However, in this case the operations with noisy and distorted data are not supported.

That is why to resolve these complexity- and privacy concerns we will, in the next section, consider a new modified protocol where the similarity strategy is replaced by the concept of exact matching even for noisy data.

3 Proposed protocol

We propose another version of the privacy preserving protocol shown in Fig. 3, where the role of the server is reduced to the confirmation of the presence of the protected query in the protected database and sending back the identification information in case of a positive confirmation. However, the simplification on the server side comes with the simultaneous increase of a number of challenges per query at the user side. The data user query consists of a number of challenges

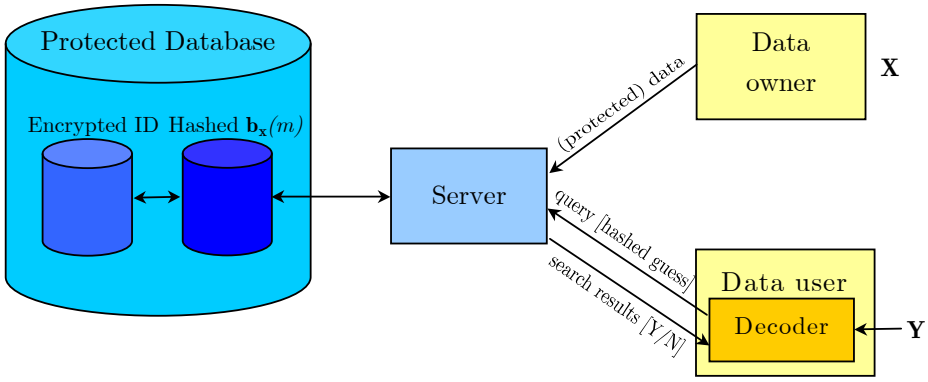

Fig. 3. Proposed data search system.

computed from $\mathbf{y}$ that are validated via the server. To illustrate the search strategy, we will refer to the binary feature space shown in Fig. 4, assuming that the data user query $\mathbf{y}$ is transformed into the binary counterpart $\mathbf{b_y}$ according to the mapping presented in Section 2.1. In the transformed binary domain, one can

assume that $\mathbf{b_y}$ has some relationship to a codeword $\mathbf{b_x}(m)$ via the considered equivalent BSC with the bit error probability $\bar{P}_b$ due to the relationship $p(\mathbf{y}|\mathbf{x})$ in the direct domain. The database stored on the server contains encrypted versions of binary templates $\mathbf{b_x}(m)$, $1 \leq m \leq M$ and the data user sends also the encrypted versions of his challenges. Observing L bits over the BSC with $\bar{P}_b$ t_b bits can be flipped. Therefore, the codeword $\mathbf{b_y}$ will be located on the distance t_b from $\mathbf{b_x}(m)$. The number of bits that can be flipped is random and T_b follows binomial distribution, i.e., $T_b \sim B(L, \bar{P}_b)$. Therefore, one can define an upper bound on the maximum number of error bits as:

$$t_{b_{\mathrm{Max}}} = B^{-1}(1 - \epsilon, L, \bar{P}_b), \tag{15}$$

where $B^{-1}(.)$ is inverse binomial cumulative density function and ϵ is a arbitrarily small chosen probability that the number of bits flipped exceeds $t_{b_{\mathrm{Max}}}$. Therefore, one search strategy consists in the sequential validation of all can-

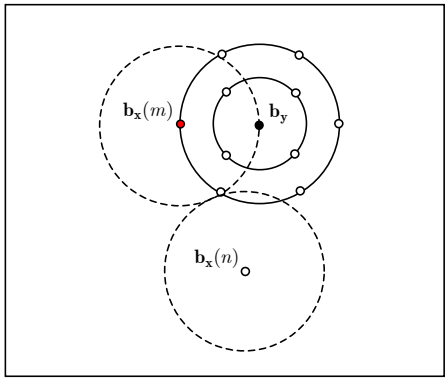

Fig. 4. Binary feature space search strategy: the algorithm checks out all possible codewords around $\mathbf{b_y}$ within the defined distance unless the right codeword $\mathbf{b_x}(m)$.

didate codewords located at the distances up to $t_{b_{\mathrm{Max}}}$ from $\mathbf{b_y}$ until an exact match with the closest codeword stored in the database is found. If there are multiple matches, a list of codewords will be returned. To get the confirmation about the data user guess, the data user encrypts its guess and sends it as a challenge to the server, which confirms or rejects the presence of the encrypted entry in the database. Obviously, such a procedure requires multiple rounds of communications and the number of guesses is:

$$\mathbf{N} = \sum_{t_b=0}^{t_{b_{\mathrm{Max}}}} \binom{L}{t_b}, \tag{16}$$

that might be prohibitively large in certain cases. It should be pointed out that the data user can also define himself the radius of the search as a free parameter.

For example, for only one error $t_{b_{\mathrm{Max}}} = 1$, one should send $O(L)$ challenges, whereas for $t_{b_{\mathrm{Max}}} = 2$, the number of challenges is additionally increased to $O(L^2)$ and for $t_{b_{\mathrm{Max}}} = 3$, even to $O(L^3)$.[4] Moreover, to reduce the communication load and to restrict any security leak to the unauthorized user, who is observing all communications, the number of challenges should be as small as possible.

Under the typicality condition for binary templates [15], we will also introduce an information-theoretic interpretation of this protocol. The mutual information between $\mathbf{B_x}$ and $\mathbf{B_y}$ is:

$$I(\mathbf{B_x}; \mathbf{B_y}) = H(\mathbf{B_x}) - H(\mathbf{B_x}|\mathbf{B_y}), \tag{17}$$

where $H(\mathbf{B_x})$ is the entropy of binary template and for independent bits the conditional entropy $H(\mathbf{B_x}|\mathbf{B_y}) = LH(B_\mathbf{x}|B_\mathbf{y})$ with $H(B_\mathbf{x}|B_\mathbf{y}) = H_2(\bar{P}_b) = -\bar{P}_b \log_2 \bar{P}_b - (1-\bar{P}_b)\log_2(1-\bar{P}_b)$ that is the binary entropy; and for equilikely bits $H(\mathbf{B_x}) = L$.

According to the weak law of large numbers, the most likely distorted codewords $\mathbf{b_y}$ will be on the radius $\bar{t}_b = L\bar{P}_b$ from $\mathbf{b_x}$ for sufficiently large L. It can also easily be confirmed that the number of these codewords will not exceed:

$$\bar{\mathbf{N}} = \binom{L}{t_b} \leq 2^{LH_2\left(\frac{t_b}{L}\right)}, \tag{18}$$

which yields $\bar{\mathbf{N}} \leq 2^{LH_2(\bar{P}_b)}$ for $\bar{t}_b = L\bar{P}_b$.

This result can be very useful for the understanding of the above protocol. It basically shows that to guess the right codeword in the properly designed codebook based on $\mathbf{b_y}$ the data user needs approximately $\bar{R} = 2^{LH_2(\bar{P}_b)}$ trials. If the query approaches the data stored in the database, $\bar{P}_b \to 0$ and $\bar{R} \to 1$. This consideration links the proposed protocol with error correction codes, where $H(\mathbf{B_x}|\mathbf{B_y}) = LH_2(\bar{P}_b)$ can be considered as the rate of the parity check bits needed to be attached to the codeword of informative bits to correct the occurred errors. At the same time, it shows how much rate is lost in $H(\mathbf{B_x})$ due to the channel imperfection. In the scope of the considered protocol, it reflects the prior imperfection or mismatch between data owner codeword $\mathbf{b_x}$ and data user query $\mathbf{b_y}$. Simultaneously, the above consideration provides an idea about the number of challenges per query that will be observed by the unauthorized user.

At the same time, the number of requests per query depends on the accuracy of $\mathbf{b_y}$ with respect to $\mathbf{b_x}$ and can be quite high for both complexity, communication and any potentially undesirable information disclosure. Therefore, to satisfy these complexity-security requirements, we propose an enhanced approach, which allows reducing the number of challenges and is based on a concept of bit reliability.

[4] One possible guessing strategy might consist in the verification of more likely error patterns that correspond to a higher probability of T_b. In this case, one can introduce a probabilistic measure of the average number of challenges versus the worst case.

4 Concept of bit reliability and challenging strategy

The bit error probability (12) depends on the magnitude of the projected coefficients. The larger the magnitude, the lower the probability of a bit error. In turn, the magnitude $\tilde{x}_i$ is determined by the angle between $\mathbf{x}$ and the projection $\mathbf{w}_i$. The closer this angle is to zero (collinear vectors), the larger magnitude and the more difficult it is for noise to change the sign of the projection $\tilde{y}_i$.

Therefore, the concept of bit reliability can be efficiently used to decrease the amount of challenges per query. Contrary to the previous guessing approach with the $t_{b_{\text{Max}}}$ number of errors, where one needs to test all possible bit error patters within $\mathbf{N}$ trials, one can assign the different bit error probabilities and localize the positions of the less reliable bits based on $Q\left(\frac{\tilde{y}_i}{\sigma_Z}\right)$ or directly observing $\tilde{y}_i$. This framework closely mimics soft decoding algorithms used in error correction codes with the only difference that the parity check bits are presented by the hash stored on the remote server [5].

One possible strategy consists in the ranking of all L bits according to their reliabilities and successive verification of all bit patterns starting with least reliable bits. This search strategy has the form of a tree and can be implemented at most with $2^{t_{b_{\text{Max}}}}$ trials. Therefore, this challenging strategy is much more efficient than the blind guessing approach discussed above.

The discussed algorithm is schematically shown in Fig. 5 and its pseudocode is presented in Algorithm 1.

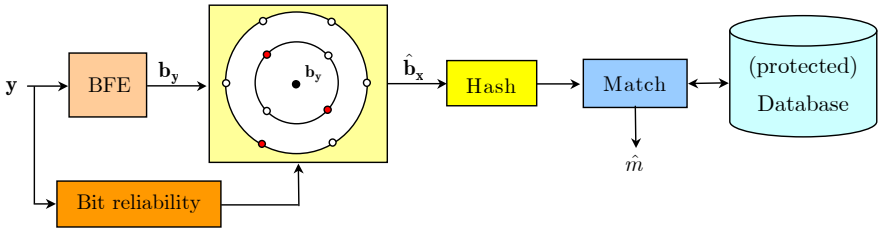

Fig. 5. Basic block-diagram of proposed bit reliability based information search system.

5 Performance-Privacy trade-off

We also consider the proposed protocol in the scope of a joint reliability- privacy analysis, where the reliability is characterized by the performance in terms of

[5] It should be pointed out that the error correction codes are structured while the templates are random that does not allow to achieve the same low complexity as for the error correction codes.

Input: Private database $\mathbf{b_x}(m)$ and vector $\mathbf{y}$
Output: Index $\hat{m}$
$Decoder(\mathbf{b_x}, \mathbf{b_y})$
Data user computes binary query $\mathbf{b_y}$ based on $\mathbf{y}$;
Data user computes bit reliability based on $\mathbf{y}$ and ranks all bits;
Data user assigns the expected number of errors $t_{b_{\mathrm{Max}}}$;
Data user assigns the first query $\hat{\mathbf{b}}_\mathbf{x} = \mathbf{b_y}$;
for $t_b = 1$ *to* $2^{t_{b_{Max}}}$ **do**
  Data user generates a hash from $\hat{\mathbf{b}}_\mathbf{x}$;
  Data user sends the hash as a query to Server;
  Server returns the confirmation;
  if *If the reply is "Found", Data user gets index* $\hat{m}$ **then**
    goto "Final decision"
  end
  if *If the reply is "Not found"* **then**
    Data user decides "No match"
  end
end
if *If the reply "Not found" is returned after* $2^{t_{b_{Max}}}$ **then**
  Data user modifies $\hat{\mathbf{b}}_\mathbf{x}$ according to bit reliability
end
Final decision;

Algorithm 1: The proposed search algorithm based on bit reliability.

the probability of error of the search. We assume that the privacy-preserving search system should satisfy:

$$\text{reliability: } \Pr[\hat{M} \neq M] \leq \delta, \tag{19}$$

$$\text{while maximizing retrieval rate} \max_{\mathbf{W},Q} \frac{1}{N} I(\mathbf{B_x}; \mathbf{B_y}) \leq \frac{1}{N} I(\mathbf{X}; \mathbf{Y}) \tag{20}$$

$$\text{privacy leak:from data owner database } I(\mathbf{X}; \mathbf{B_x}) \leq N(L_{p_x} + \delta), \tag{21}$$

$$\text{under} \max H(\mathbf{B_x}) \tag{22}$$

$$\text{from data user query } I(\mathbf{X}; \mathbf{B_y}) \leq N(L_{p_y} + \delta), \tag{23}$$

with small nonnegative δ where Q denotes the binarization.

The generalized representation of proposed system is shown in Fig. 6, The blocks of dimensionality reduction based on random projections $\mathbf{W}$ and binarization Q correspond to (4) and (8), respectively, and form the binary template extraction. The block PA denotes privacy amplification that will be considered below. In the considered sense, the better privacy is achieved by the reduction of privacy leaks L_{p_x} and L_{p_y} that is achieved by the randomization of the outputs. In turn the binary template extraction and privacy amplification sequentially reduce the rate $\frac{1}{N} I(\mathbf{X}; \mathbf{Y}) \geq \frac{1}{N} I(\tilde{\mathbf{X}}; \tilde{\mathbf{Y}}) \geq \frac{1}{N} I(\mathbf{B_x}; \mathbf{B_y}) \geq \frac{1}{N} I(\mathbf{B'_x}; \mathbf{B'_y})$ [14]. Therefore, one should carefully address this performance-privacy trade-off.

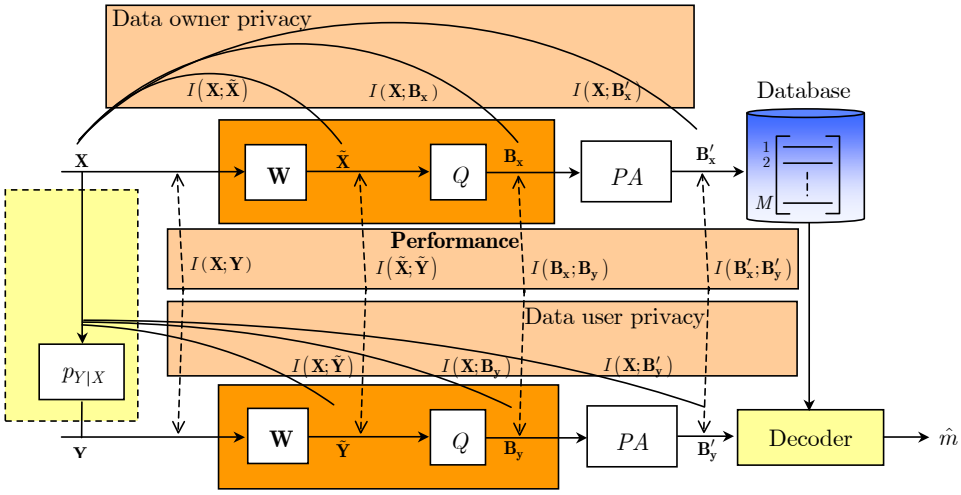

Fig. 6. Generalized performance-privacy diagram.

One of the critical issues in the privacy enhancement is the minimization of the leak from the data owner's database $I(\mathbf{X};\mathbf{B'_x})$ whilst simultaneously maximizing the entropy $H(\mathbf{B'_x})$ of the template.

The minimization of the leak $I(\mathbf{X};\mathbf{B'_x})$ should be performed under the assumption that the data owner provides a sufficient amount of information in $\mathbf{B'_x}$ about $\mathbf{X}$ to perform a unique search according to the reliability constraint. At the same time, an unauthorized user should not be able to reconstruct $\hat{\mathbf{X}}$ based on $\mathbf{B'_x}$ with sufficient accuracy to compromise the anonymity or to extract some information that is irrelevant in the scope of the unique search problem.

The maximization of $H(\mathbf{B'_x})$ is important for several reasons. First of all, maximizing the entropy $H(\mathbf{B'_x})$ by producing a uniform string from the generic data $\mathbf{X}$ (or $\mathbf{B_x}$) is important from a cryptographic point of view that is analog to key extraction that enables the use of cryptographic primitives at the later stages. Secondly, it also plays a quite important role for efficient data storage because uniform data is the most efficient in terms of entropy source coding for large databases. Thirdly, using uniform data represents an efficient form of data search avoiding issues related to the curse of dimensionality. Therefore, the problem we consider next is the conversion of a generic $\mathbf{X}$ into the uniform $\mathbf{B'_x}$. We will refer to this problem as a *generic privacy amplification* (GPA).

In the scope of the proposed protocol, one can implement GPA in two different ways. The first approach consists in the diagonalization of the covariance matrix $\tilde{\mathbf{K}}_X$ of the transformed data $\tilde{\mathbf{X}}$ while the second one refers to the privacy amplification based on universal hashing.

The diagonalization based GPA is schematically presented in Fig. 7. Under the proper selection of a random projection matrix with Gaussian basis vectors

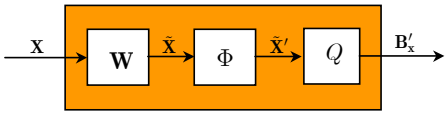

Fig. 7. Diagonalization based GPA.

one can expect that the projected vector will follow a Gaussian distribution $\tilde{\mathbf{X}} \sim \mathcal{N}(\mathbf{0}, \tilde{\mathbf{K}}_X)$ where the covariance matrix $\tilde{\mathbf{K}}_X$ that can easily be estimated. However, it does not guarantee that $\tilde{\mathbf{K}}_X$ will be diagonal. To achieve this goal, we apply a diagonalization operator Φ, which is designed based on the principle component analysis (PCA) transform, i.e., its basis vectors correspond to the eigenvectors of $\tilde{\mathbf{K}}_X$. As a result, one can obtain uncorrelated random variables in the vector $\tilde{\mathbf{X}}$ in general and independent components for Gaussian distributions.

The cryptographic privacy amplification was proposed in [16] based on universal hash functions [17] and extended in [18]. In the scope of this sort of GPA, the privacy amplification component consists in a mapping of nonuniform random template $\mathbf{B_x}$ to a shorter, almost uniform string $\mathbf{B'_x}$. In principle, one can use different operations to achieve this goal: (a) extracting L' random bits from $\mathbf{B_x}$; (b) computing L' arbitrary parity checks of $\mathbf{B_x}$; (c) arbitrary random function mapping L-bit $\mathbf{B_x}$ to L'-bit $\mathbf{B'_x}$; (d) transmitting $\mathbf{B_x}$ via a BSC with a bit error probability P_{PA} satisfying $L' = L - LH_2(P_{PA})$; (e) securely encoding the ID and embedding it as a random watermark $\mathbf{B_w}$ into $\mathbf{B_x}$ using coding with side information, which is well established in digital watermarking. In data mining, (d) corresponds to additive randomization and (a) corresponds to multiplicative (dimensionality reduction) randomization [13].

The main idea behind universal hashing is to define a collection $\mathbb{G}$ of hash function in such a way that a random choice of a function $g \in \mathbb{G}$ yields a low probability that any two distinct inputs $\mathbf{b_x}(m)$ and $\mathbf{b_x}(n)$ will collide when their hashed values are computed using a function g. The larger the cardinality of this class $|\mathbb{G}|$, the lower this probability for all choices of $\mathbf{b_x}(m)$ and $\mathbf{b_x}(n)$. However, for the purpose of practical implementation, it is important not only to have small probability of collision, but $|\mathbb{G}|$ should be small as well. This is because $\log_2 |\mathbb{G}|$ random bits are required to specify a choice of a random hash function in the class $\mathbb{G}$. We refer the readers to [20] for more details about the low complexity implementation of these functions.

6 Experimental results

To demonstrate the validity of the prosed approach, we performed a number of simulations on synthetic Gaussian data and on real images. The tests have been performed under various distortion models including additive white Gaussian, uniform noise and lossy JPEG compression. The test data base consists of $M = 1'000'000$ entries. We only use $N = 32 \times 32$ blocks for each image for simulation

purposes. A binary feature vector of length $L = 32$ is extracted from each block and stored in a properly hashed database.

The implementation of the proposed search algorithm is based on the branch and bound technique, where the least reliable bits are first candidates for flipping. The depth of the search tree has been limited to 12, and can otherwise be limited even further if sufficiently few bits will be flipped due to the noise, depending on the SNR as described in Section 3.

The simulation consists of two parts. First, we investigate the bit error probability for the binary templates under AWGN, uniform noise and lossy JPEG compression. This provides an idea about the corresponding amount of errors and necessary number of trials per query. Second, we implement the described protocol and measure the accuracy of the search according to the average probability of error. All results have been obtained for 100 noise and random projection matrix realizations.

The bit error probabilities for the AWGN, additive uniform noise and lossy JPEG compression are shown in Fig. 8, Fig. 9 and Fig. 10, respectively. The query model is considered in terms of the signal-to-noise ratio (SNR) defined as $\mathrm{SNR} = 10\log_{10}\frac{\sigma_X^2}{\sigma_Z^2}$ for additive noises and in terms of quality factor for compression. It is done with the purpose to reflect the incompleteness of data user knowledge and mismatch with the stored database.

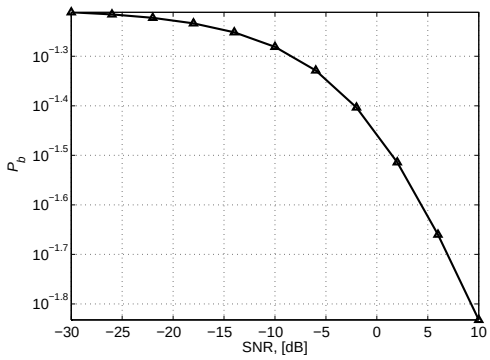

Fig. 8. Bit error probability for AWGN.

In this paper, we tested the search system in the identification mode, when only one match is requested. Multiple matches are considered as incorrect identification. The probability of incorrect identification was tested for databases containing 8K (2^{13}) and 1M (2^{20}) entries under AWGN using binary templates of length $L = 32$. The reason for selecting the AWGN channel is that it produces the highest bit error rate amongst the considered channels, so it represents a worst case. The probability of search error is shown in Fig. 11. As we have

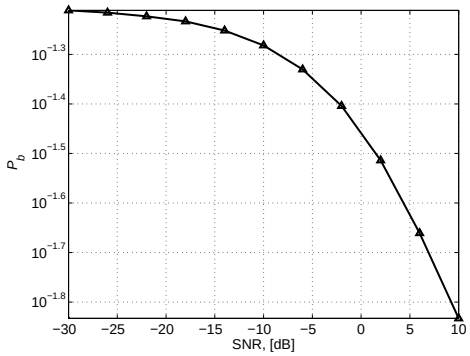

Fig. 9. Bit error probability for additive uniform noise.

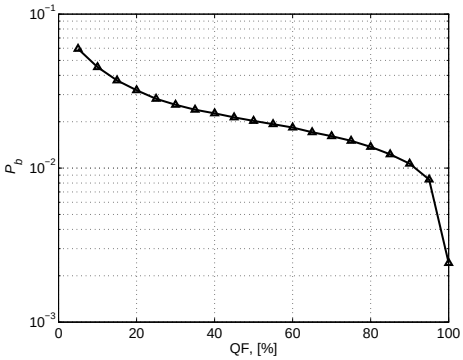

Fig. 10. Bit error probability for lossy JPEG compression.

pointed out, the described system can also return multiple matches according to the list decoding mode that will be presented in the final version of the paper upon acceptance. Especially the results for the database of 1M elements are very interesting. Usually, one would expect to observe the behavior that is shown by the simulations on a database of 8K: the exhaustive search using the minimum Hamming metric should be optimal, whereas the proposed method only tests a subset of the database and should therefore at best approximate the result of the exhaustive search. However, the results for the larger database show exactly the inverse results: the approximative method outperforms the exhaustive search! These seemingly unlikely results can be explained by the following arguments: due to the large size M of the database relative to the length of the templates L, there could frequently be several codewords that have the same Hamming distance to the channel output. From these equally good candidates, the exhaustive

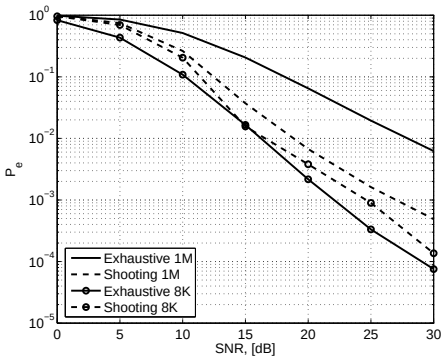

Fig. 11. Average probability of search error for AWGN.

search will simply select the first one of the best entries it comes across, being the one that has the lowest index in the database. The proposed method on the other hand, will select the first one it comes across whilst traversing its search tree. The order of the walk through the tree is determined by the reliability of the bits, leading to the effect that the algorithm will first hit the entry with the bit that were most likely to flip. Thus, it will prefer the entries where bit-flips occurred at positions that were deemed unreliable, rather than whichever entry happens to occur first in the codebook. The improvement over the exhaustive search can thus be explained by the use of reliability information in the shooting approach.

7 Conclusions

In this paper, we attempted to present an alternative signal processing view on the privacy preserving search problem. The proposed approach mimics the work of error correction codes with soft information about bit reliability implemented in the distributed manner. The results of computer simulations prove the consistency of the proposed framework for the broad class of distortions models from the signal processing family of distortions. In our future research, we will also concentrate on the extension of our results to the broader family of distortions including geometrical transformations as well as propose even more efficient search strategies that are currently under testing for large databases.

In the scope of our envisioned extensions, we will investigate the broader class of distortions under both unique and list decoding regimes. We will also perform the practical evaluation of privacy leakages based on PCA and ICA analysis.

8 ACKNOWLEDGMENTS

This work is supported by SNF projects 111643 and 1119770.

References

1. Brinkman, R.,: Searching in encrypted data. PhD. thesis, University of Twent, Twente, The Netherlands (2007)
2. Li C., HacGumus H, Iyver B. and Mehrota S.: SSQL: Secure SQL in an insecure environment. VLDB journal, 2006.
3. Iyver B., HacGumus H and Mehrota S.: Efficient execution of aggregation queries over encrypted relational databases. In Proceedings of the 9th International Conference on Database Systems for Advanced Applications, March 2004.
4. Iyver B., HacGumus H. and Mehrota S.: Efficient execution of aggregation queries over encrypted relational databases. In Kyu-Young Whang Yoon Joon Lee, Jianzhong Li and Doheon Lee, editors, Database systems for Advanced Applications: 9th International Conference, DASFAA, volume LNCS 2973, pages 125–136. Springer Verlag, Berlin, March 2004.
5. Agrawal R., Kieman J., Srikant R. and Xu Y.: Order-preserving encryption for numeric data. In Proceedings of the ACM SIGMOD 2004 Conference, June 2004.
6. Boneh D., Crescenzo G., Ostrovsky R. and Persiano G. Public-key encryption with keyword search. In C. Cachin, editor, Proceedings of Eurocrypt 2004, 2004.
7. Song D. X., Wagner D. and Perrig A.: Practical techniques for searches on encrypted data. In IEEE Symposium on Security and Privacy, pages 44–55, 2000.
8. Kushilevitz E. and Ostrovsky R.: Replication is not needed: single database, computationally-private information retrieval. In IEEE Symposium on Foundations of Computer Science, 364-373, 1997.
9. Chor B. and Gilboa N.: Computationally private information retrieval (extended abstract). In Proceedings of the 29th ACM Symposium on the Theory of Computing, pages 304–313, El Paso, Texas, United States, 1997.
10. Chor B., Kushilevitz E., Goldreich O. and Sudan M.: Private information retrieval. Journal of the ACM, 45(6):965–981, November 1998.
11. J. Domingo-Ferrer. A new privacy homomorphism and applications. Information Processing Letters, 22(6):644–654, November 1996.
12. SPEED Deliverable D3.2: Identification of Requirements and Constraints, (2007), `http://www.speedproject.eu/index.php?option=com_docman&task=cat_view&gid=45&Itemid=37`
13. Privacy-Preserving Data Mining Models and Algorithms Series: Advances in Database Systems , Vol. 34 Aggarwal Charu C.; Yu Philip S. (Eds.) 2008.
14. Voloshynovskiy S., Koval O., Beekhof F. and Pun T.: Conception and limits of robust perceptual hashing: toward side information assisted hash functions. In: Proceedings of SPIE Photonics West, Electronic Imaging / Media Forensics and Security XI, San Jose, USA (2009)
15. Cover. T., Thomas. J.: Elements of Information Theory. Wiley and Sons, New York (1991)
16. Bennett, C.H., Brassard, G. and J.-M. Robert: How to reduce your enemy's information, In Proc. Advances on Cryptography Crypto'85, Lecture Notes in Computer Science, Volume 128, Springer-Verlag, Berlin, 468–476, 1986.
17. Carter J.L. and Wegman M.N.:. Universal classes of hash functions. Journal of Computer and System Sciences, Volume 18, Issue 2, 143–154, 1979.
18. Bennett, C.H., Brassard, G., Crepeau, C. and Maurer, U.M.: Generalized privacy amplification. IEEE Transactions on Information Theory, Volume 41, Issue 6, 1915–1923, 1995.

19. Stinson D.R.: Universal hashing and authentication codes. Designs, Codes, and Cryptography, 4, 369–380, 1994.
20. Kaps J.-P., Yuksel K. and Sunar B.: Energy Scalable Universal Hashing, IEEE Transactions on Computers, Volume 54, Issue 12, 1484–1495, December, 2005.