Robust Geotag Generation Algorithms from Noisy Loran Data for Security Applications

Di Qiu, Sherman Lo, Per Enge, Dan Boneh, *Stanford University*

BIOGRAPHY

Di Qiu is a Ph.D. candidate in Aeronautics and Astronautics working in the Global Positioning System (GPS) Laboratory at Stanford University. Her current research interests are location-based security, signal authentication, information theory, and fuzzy extractors.

Dr. Sherman Lo is currently a research associate in the Stanford University Global Positioning System (GPS) Laboratory. He is the Associate Investigator for the Stanford University efforts on the Department of Transportation's technical evaluation of Loran.

Per Enge is a Professor of Aeronautics and Astronautics at Stanford University, where he is the Kleiner-Perkins, Mayfield, Sequoia Capital Professor in the School of Engineering. He directs the Stanford GPS Research Laboratory.

Dan Boneh is an associate professor of Computer Science and Electrical Engineering at Stanford University. He is a well-known researcher in the areas of applied cryptography and computer security.

ABSTRACT

Geo-security, or location-based security service, provides authorization of persons or facilities based on their distinctive location information. It applies the field of position navigation and time (PNT) to the provision of security. Location-dependent parameters from radio navigation signals are quantized to compute a location verification tag or “geotag” to block or allow accesses by users. Adequate quantization steps of location-dependent parameters should be selected to achieve reliable performance.

Loran is chosen as a case study because of its beneficial properties for location-based security services. The achievable performance and security of the system are determined by the quantity and quality of location-dependent parameters. By quantity, we mean the total number of different (independent) location-dependent measurements available. By quality, we mean the amount of unique location-dependent information and its consistency provided by each parameter that can be used to generate a robust geotag. It is desirable that the parameters be relatively insensitive to temporal changes that can weaken the uniqueness of the information. As a result, reproducibility and repeatable accuracy are fundamental requirements for any location-based security service. In practice, quantization temporal variations in location-dependent parameters significantly degrade system reliability.

In this paper we introduce two new methods to generate strong geotags from noisy location data: fuzzy extractor-based and classifier-based. The performance of the different geotag generation algorithms are analyzed and compared; real data are applied to evaluate Loran getoag reliability and spatial discrimination.

INTRODUCTION

In this paper we introduce a security-oriented location-based service and use Loran as a case study. In general, location-based services require accurate estimation of position, e.g., latitude, longitude, and altitude, from location measurements. We show that for a number of security applications there is no need to map location measurements into an accurate global position. Loran, which operates in most of the northern hemisphere, has many advantages over satellite-based navigation systems for secure location-based service. It is a high-power terrestrial signal that easily penetrates buildings and cities where line-of-sight signals are not feasible. The stationary transmitters can result in many parameters that are solely location-dependent, but not time-dependent. The location-dependent parameters have high repeatable accuracy, which is essential to the robustness of derived geotags. In addition, the modernized Loran or eLoran has a data channel that not only improves navigation performance but benefits the geo-security design. The Loran location-based parameters are used to derive a geotag, which is a piece of information that allows or restricts access for security applications. We provide examples of location-based security applications in two categories: block-listing and white-listing.

- *Block-listing*: An example of a block-listing application is digital manners policy (DMP). Technologies for DMP [1] attempt to enforce manners at public locations. A DMP-enabled cell phone can be programmed by the cellular service provider to turn off the phone's camera while inside a hospital, locker room, or classified installation. Or the phone can be programmed to switch to vibrate mode while inside a movie theater. Although some of these ideas maybe highly controversial [2], in this paper we focus only on the technical aspects of the application. Using our geotag, one can build a list of geotags where the camera will be turned off. The device downloads an updated list periodically, and when the device encounters a geotag in the blocklist, it turns the camera off. When the device leaves the blocked location, the camera is turned back on. Thus, digital manners are enforced without telling the device its precise location.

- *White-listing*: An example of white-listing is location-based access control. Consider a location-aware disk drive: the drive can be programmed to work only while in the secure data center; an attacker who steals the device will not be able to interact with it. Location-based access control using encryption was studied by Scott and Denning [3] under the name Geoencryption. Another white-listing application is Loopt, which provides geo-social networking services to users, enabling them to locate friends via their GPS-based cell phones. To implement Loopt, a central server is required to compute geotags, perform matching algorithms, and notify users with SMS messages if they and their friends are in a given location. Since the computed geotags cannot reveal users' location information, the users' privacy can be protected.

A location-based security system must survive the following attack: the attacker owns the device and tries to make the device think it is somewhere else. To defend against this threat, we make two assumptions. First, a device that integrates a location sensor and geotag generation algorithm should be tamper-resistant. If the device is not tamper-resistant, it can, be attacked, for example, by replacing the received location parameters with fake ones; by brute force attack; or by tampering with the tag database. Second, the radio signal is self-authenticated to allow users to verify the source of incoming signals. A signal authentication protocol, Timed Efficient Stream Loss-tolerant Authentication (TESLA), is proposed on Loran. We propose a means of implementing TESLA for authentication on navigation signals. The implementation was tested on a West Coast Loran station in January, 2007 [4]. The theoretical analysis and experimental results of TESLA authentication performance were discussed previously in [5].

Additionally, it is desirable that geotags are reproducible; thus, location-dependent parameters should be relatively insensitive to temporal changes. Reproducibility means that measurements at the same location at different times will always produce the same tag. Reproducibility is a fundamental requirement to derive a robust geotag. However, several types of errors presented in the radio frequency (RF) signals can degrade the performance of location-based security service. This paper presents two new geotag generation algorithms. The first method uses fuzzy extractor to improve the geotag reproducibility. The second method applies a pattern classification technique and develops classifier-based algorithms to generate strong geotags from noisy location data. These geotag constructions can also be applied to other RF signals, such as satellite-based, Wi-Fi, DTV, and cellular signals, and non-RF signals such as infrared and ultrasound.

The structure of the paper is organized as follows: We first describe system models of a location-based security system and the error patterns of location-dependent parameters. This paper then defines fuzzy extractor, shows three different constructions of fuzzy extractors, and evaluates the geotag reproducibility based on the fuzzy extractor constructions. We then provide a short review of pattern classification and classifiers. Three different constructions of classifier-based geotags will be introduced. We evaluate the spatial discrimination of computed geotags based on the classifiers in the subsequent sections. This paper then summarizes and concludes with future directions of the research.

SYSTEM MODELS

Reproducibility and repeatable accuracy are desirable qualities in location-based security systems. They allow a user to provide location-dependent parameters for the derived tag at calibration, and preserve the validity of the parameters at a later time for verification.

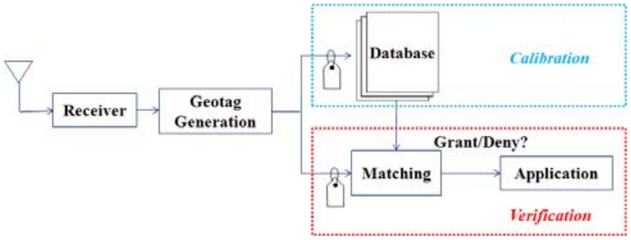


Figure 1. Location-based security system

Figure 1 illustrates how the system works. Location-dependent parameters from the surveyed locations are

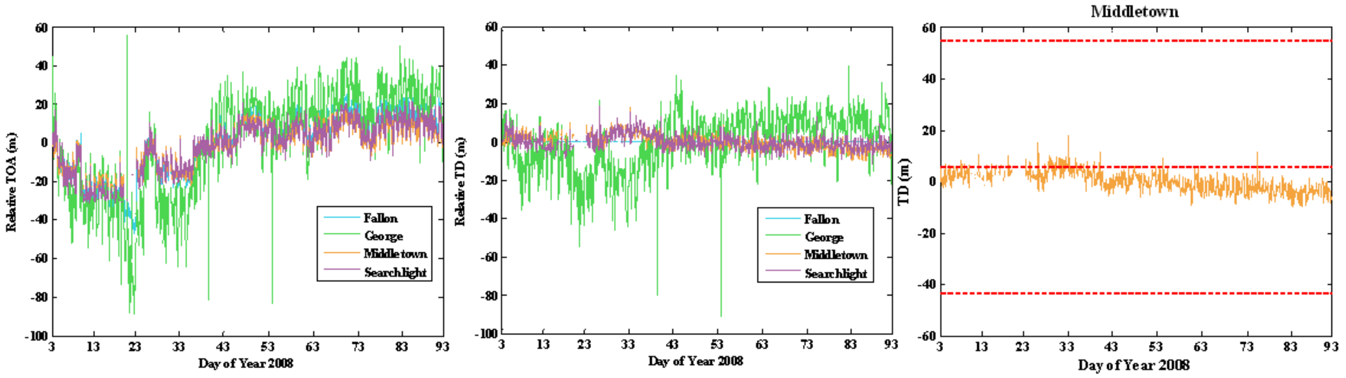


Figure 2. TOA with zero means (left); TD with zero means (middle); TD quantization (right)

mapped into tags and stored in a central database in the calibration step. At verification, the user matches his computed tag with the stored tag to validate the correctness of the user's location.

The signal characteristics should be sufficiently consistent that when the user is ready to verify, measurements at the same location will yield the same previously generated tag. Temporal variation reflects the instability or degree of scatter within a particular parameter at a given location, and increases the likelihood of mismatched tags. The current geotag generation consists of three steps: extracting *features* or location-based parameters from the received location signals, quantizing the parameters with chosen step sizes, and mapping the quantized parameters into a binary string. The binary mapping process can be done using a hash function, which is easy to compute, but difficult to invert.

Performance Metrics

The problem of deciding whether or not the computed geotag is authentic can be viewed as a hypothesis-testing problem. The task is to decide which of the two hypotheses, H_0 (accepting as an authorized user) or H_1 (rejecting as an attacker), is true for an observed location measurement. A location-based security system can make two types of errors: 1) mistaking measurements from the same location as from two different locations, and accepting hypothesis H_1 when H_0 is true (called a *false reject*); and 2) mistaking the measurements from two different locations as from the same location and accepting H_0 when H_1 is true (called a *false accept*). Both *false reject rate* (FRR) and *false accept rate* (FAR) depend on the accuracy of the Loran receiver and the quantization step chosen to quantize location parameters. FAR only applies to white-listing applications, while FRR can be a performance metric for both block-listing and white-listing applications.

FRR and FAR can be traded off against each other by varying the quantization step size. A more secure system aims for low FARs at the expense of high FRRs, while a more convenient system aims for low FRRs at the expense of high FARs.

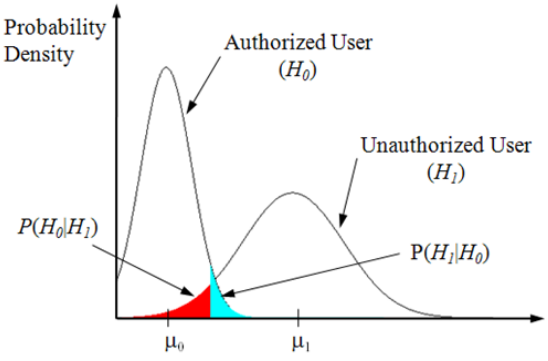


Figure 3. Performance metrics: false reject rate and false accept rate

Types of Errors

First, we study the various types of errors presented in location data to achieve optimal generation of geotags. The most common error source is thermal and atmospheric noise. Thermal noise, considered as white Gaussian, cannot be eliminated and always presents in all electronic devices and transmission media. Loran atmospheric noise, caused by lightning, is non-Gaussian and dominant in low-frequency signals, and can be impulsive if the lightning is local. Both thermal and atmospheric noises depend on the propagation path, the distance between transmitter and receiver, the quality of the receiver, and the local noise floor, etc.

Another error source is bias. An example of seasonal bias in Loran signals is Additional Secondary Factor (ASF), which is the additional delay in propagation time due to the signals traveling over a mixed path: e.g., seawater and land with various conductivities. This error introduces large seasonal variations in time-of-arrival (TOA), as shown on the left of Figure 2. The four stations, Fallon, George, Middletown and Searchlight, are from the Loran West Coast chain, Group Repetition Interval (GRI) 9940. Fallon is the master station of GRI 9940, while the remaining three are the secondary stations. The monitor data were collected at Stanford University for a 90-day period to observe seasonal variations in Loran signals. The delay can be significant and can introduce a position error of hundreds of meters [6]. Thus ASF represents one of the largest error sources in Loran. Many factors affect ASF, including soil conductivity, temperature, humidity, local weather, etc. Therefore, ASF varies both temporally and spatially. This raises the difficulty of modeling ASF over CONUS. The temporal component derives from all of the time-varying aspects, while the spatial component takes into account the non-uniform ground conductivity and topography [7]. Many methodologies have been developed to mitigate ASF. In the previous study [8], we demonstrated two simple ideas: time difference and “previous day is today’s correction.” Time difference (TD) is the difference in TOAs between secondary stations and the master station; thus, the master station is used as a reference to remove the ASF bias. The second method is to use the previous day’s ASF measurements as today’s correction. This requires that either the user receiver constantly monitors Loran data, or a reference station that is near the user broadcasts the previous day’s ASF as a correction via a data channel. Neither method removes ASF completely. The TD method has spatial decorrelation due to the different propagation paths of master and secondary stations. The previous day’s correction suffers from the temporal decorrelation of ASF, because the previous day’s ASF is different from today’s ASF. In this paper we use the TD method to mitigate partial ASF temporal variations, because it corrects more ASF biases, per our previous study [8]. The TD measurements from four stations are plotted in the middle of Figure 2.

In addition, quantization error, which is the difference between the value of a continuous parameter and its quantized value, can cause the system to fail to reproduce a correct geotag. The quantization error is usually correlated with the thermal noise, the atmospheric noise, and the seasonal biases discussed above. We cannot guarantee that the measurements are always in the middle of the quantization grid. In the worst case, the measurements lie on the boundary of the grid, as illustrated in the right plot of Figure 2. The figure plots the TD measurements from Middletown with zero mean. The red dash lines represent the quantization grid boundaries. Even though the quantization step is chosen to overbound signal variations due to random noise and seasonal biases, the quantization error increases the likelihood of failure to reproduce a geotag. Figure 4 depicts the false reject rate as a function of the quantization step in terms of the parameter standard deviation for the computed Loran geotag, which is derived from TD, ECD, and SNR of the four West Coast stations. The curve does not monotonically decrease as the quantization step increases. The non-monotonic relationship is a result of the quantization error.

The above error types are considered to be Euclidean metric. The last type of error is Hamming metric and comes from the operations of RF systems; for example, Loran stations might be offline due to maintenance or other implementation issues.

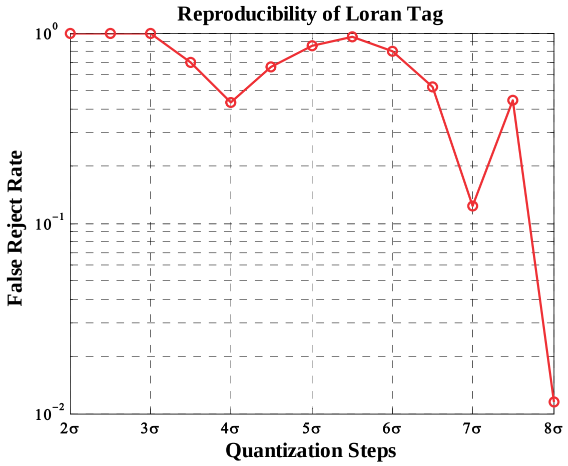


Figure 4. Loran geotag reproducibility

FUZZY EXTRACTORS

Background and Definitions

The first approach of fuzzy extractor or error-tolerant cryptographic algorithm, called fuzzy commitment scheme, is proposed for biometrics by Juels and Wattenberg [9]. The scheme uses an error correcting code to handle Hamming distance. More approaches for Hamming distance, set difference, and edit distance are introduced in [10]. It also introduces a different error tolerant algorithm, called secure sketch.

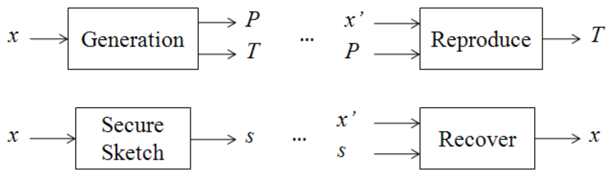


Figure 5. Fuzzy extractor (top); Secure sketch (bottom)

In this paper we follow the definition of fuzzy extractors in [10]. A fuzzy extractor works in two steps, illustrated in Figure 5. During calibration step, one runs an algorithm Gen in input $x \in M$ to generate a public value P and a geotag T, where M is a metric space of x. The public value P is stored for future use. An algorithm Rep is used to reproduce the tag *T* using *P* from noisy location vector *x’*. Fuzzy extractors are information-theoretically secure, thus we can use them for security applications without introducing additional assumptions [10]. A secure sketch also consists of two steps. A procedure SS produces *s*, called a sketch, using input *x*. Then given *s* and *x’* close to *x*, a procedure Rec can recover *x*. The sketch should not reveal much information about *x*. Unlike fuzzy extractors, a secure sketch recovers the original input *x* from noise while a fuzzy extractor reproduces geotag *T* from noisy input.

Definition 1. A fuzzy extractor is a tuple (*M*, *t,* Gen, Rep), where *M* is the metric space with a distance function dis, Gen is a generate procedure and Rep is a reproduce procedure, which has the following properties:

If Gen(*x*) outputs (*T*, *P*), then Rep(*x*’, *P*) = *T*, whenever dis(*x*, *x*’) ≤ *t*. If dis(*x*, *x*’) > *t*, then there is no guarantee *T* will be outputted.

Definition 2. A secure sketch is a tuple (*M*, *t*, SS, Rec), where *M* is the metric space with a distance function dis, SS is a sketch generating procedure and Rec is a recover procedure, which has the following properties.

Rec(*x*’, SS(*x*)) = *x*, if dis(*x*, *x*’) ≤ t. The sketch s is to be made public. We say the scheme is *m*-secure and the entropy loss of *s* is at most *m*. H(*x*) – H(*x*|*s*) ≤ *m*. H denotes the entropy of a random variable.

In this paper we propose three fuzzy extractors based on Euclidean and Hamming metrics for inconsistent location parameters.

Euclidean Metric Fuzzy Extractor

Let location vectors be *n*-dimensional in metric space *M*. We consider the distance measure for location-based parameters is L$_\infty$ norm to be conservative. We normalized the measure using Δ, and the distance is defined as

$$dis(x,x') = \left(\max_i \frac{|x_i - x_i'|}{\Delta_i} \right)_{i=1}^{n} . \qquad (1)$$

The basic idea of this fuzzy extractor is to adjust the offsets between the continuous parameters and the discrete ones after quantization. The construction of the fuzzy extractor is shown as follows

$$\mathrm{Gen}(x) = \begin{pmatrix} T = hash(q_x) \\ P = \left(x_i - \Delta_i \left\lfloor \frac{x_i}{\Delta_i} \right\rfloor \right)_{i=1}^{n} \end{pmatrix}, \qquad (2)$$

$$\mathrm{Rep}(q_x', P) = \begin{pmatrix} q_x' = \left\lfloor \frac{x'_i - P_i + \frac{\Delta_i}{2}}{\Delta_i} \right\rfloor_{i=1}^{n} \\ T' = hash(q_x') \end{pmatrix} . \qquad (3)$$

If $dis(x,x') < \frac{1}{2}$, then quantized location vector q_x’ can be reproduced, that is, *T*’ = *T*. This claim defines the reproducibility of geotag. The quantization step Δ is a design parameter. The bigger the step, the more errors can be tolerated using this fuzzy extractor.

Shannon entropy is used to measure entropy loss of fuzzy extractors mathematically. We estimate the entropy loss or the mutual information between the conditional H(*x*|*P*) and unconditional H(*x*) entropies. They are statistically independent if the mutual information is zero. Given *x* = q_x + *P*, let x’ = q_x + *P* – *δ,* where *δ* is the Euclidean difference between *x* and *x*’ due to noises and biases. Our objective is to determine an upper bound on H(*x*|*P*). By using the definition of conditional entropy [11], we obtain

$$H(x \mid P) = H(x) - H(\delta) . \qquad (4)$$

Thus, the entropy loss of public value *P* is H(*δ*). It depends on the probability distribution of *x* and the quantization step Δ. For the case *n* number of different location parameters, the total information leakage is

$$H(\delta) \le \sum_{i=1}^{n} \log(\Delta_i) . \qquad (5)$$

This equation assumes the location parameters are uniformly and independently distributed and provides an upper bound on the entropy loss. In practice, the entropy loss is small in comparison with *H*(*x*). The measured entropy in a geotag also quantifies the amount of uncertainty from an attacker’s point of view. The entropy in a geotag computed from quantized parameters is equal to *H*(q_x|*P*). By the definition of q_x, q_x is independent of *P*; thus, *P* does not leak any information on q_x. Intuitively, this makes sense that knowing the offsets between x and Δ_x q_x, one cannot figure out the user’s quantization level exactly without further information.

Reed-Solomon Based Fuzzy Extractor

The approach achieves robustness against noises and biases by making use of error-correcting codes to recover changes measured by Hamming distance. Hamming distance, defined in Equation (6), measures the number of different elements between two strings or vectors. In addition, this fuzzy extractor deals with the problem caused by offline transmitters. Geotag can be reproduced even when there are missing parameters.

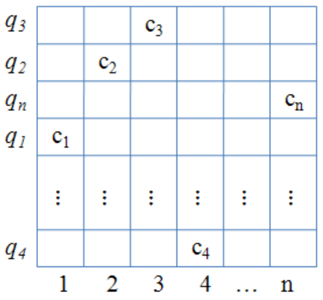
$$dis(x,x') = \sum_{i=1}^{n} x_i \oplus x'_i \qquad (6)$$

We use Reed-Solomon (RS) error-correcting code to construct a fuzzy extractor to recover the changes of the quantized location parameters. Reed-Solomon coding is a well known forward error correction coding method that potentially for burst errors [12]. The key idea of the construction is to first create a polynomial by encoding the secrets, which is the tag in location-based security system. The next step is to project the quantized location parameters on the polynomial and randomly create chaff points to hide the polynomial. At last, the secrets can be recovered from the chaff points with adequate location parameters. The detailed construction is described as follows.

Calibration. Given $q_x = \{q_1,\ldots,q_n\}$,

1. A secret message is computed from a random generator.
2. The secret message can be hashed to get a geotag *T*.
3. The geotag *T* is encoded to a vector *c* using Reed-Solomon code. The vector *c* has a size of *n*. The RS encoder (*n, k*) is chosen based on design criteria that the total number of errors *t* can be corrected is determined by (*n-k*)/2.
4. Construct mapping matrix or public information *P*. *P* has a size of *N*×*n*, where *N* is the number of quantization levels of location parameters and determined by chosen quantization steps. For each column of *P*, locate the element of vector *c* based on each quantized location parameter. For instance, if q_i = 20, then *P*(20, *i*) = c_i. Figure 6 illustrate the formation of mapping matrix *P*. Populate the rest of the matrix using random numbers. This mapping matrix is then saved for future use.

$$\text{Gen}(q_x, m) = \begin{pmatrix} T = rand \\ c = \text{RS encode}(T) \\ P = mapping(c, q_x, T) \end{pmatrix} \qquad (7)$$



q_3			c_3			
q_2		c_2				
q_n						c_n
q_1	c_1					
	⋮	⋮	⋮	⋮	⋮	⋮
q_4				c_4		
	1	2	3	4	…	n

Figure 6. Mapping matrix construction

Verification. Given q_x*'* is a location parameter vector that has *t* or less than *t* elements different from q_x.

1. Given the mapping matrix *P* generated previously.
2. Obtain a vector *c'* using *P* and q_x*'*. If q_x*'* and q_x are identical, *c'* has the same elements as *c*. If attackers have no information on location parameters q_x, it is difficult to guess a vector *c'* that satisfies dis(*c, c'*) ≤ *t* due to the large search space of mapping matrix. Such a search is equivalent to brute-force attacks.
3. Apply Reed-Solomon decode to compute *T'* from *c'*. If dis(*c, c'*) ≤ *t*, the secret message can be recovered correctly; otherwise, the output would not be the same as *T*.

$$\text{Rep}(q_x', P) = \begin{pmatrix} c' = mapping^{-1}(P, q_x') \\ T' = \text{RS decode}(c') \end{pmatrix} \qquad (8)$$

This approach makes use of the property of Reed-Solomon codes to tolerant *t* errors in the quantized location parameters. It is not fault-detective since users would not be able to find out whether the errors in received location parameters can be tolerated or not until computation of the geotag. The entropy loss of this construction is *t*log*N*. This results in the effective tag length is (*n-t*)log*N*. Thus, Hamming metric fuzzy extractors improve geotags' reproducibility at the expense of their entropy.

Secret Sharing Based Fuzzy Extractor

The third construction of fuzzy extractor is based on the idea of secret sharing. The scheme is a method of sharing secret *S* among a set of *n* participants. For any subset of *k* (*k* ≤ *n*) participants, the secret *S* can be reconstructed. But a subset of less than *m* participants will fail to reconstruct *S*.

The distance metric in this construction is also Hamming. The input to the fuzzy extractor is quantized location vector q_x. The first step of construction is to create a polynomial *f*(*x*), such that *f*(*i*) = q_i, ∀ *i* = 1, 2, …, *n*. The generation and reproduction procedures are as follows

$$\text{Gen}(x) = \begin{pmatrix} f(i) = T + a_1 x + a_2 x^2 + \ldots + a_k x^k, \\ \text{where } a_1, a_2, \ldots, a_k \text{ are random numbers} \\ P = < i_1, \ldots, i_k >, s.t.\ f(i_j) = q_j \end{pmatrix}, (910)$$

$$\text{Rep}(x', P) = \begin{pmatrix} \text{Reconstruct } f(i) \text{ using } P \text{ and } q'_x \\ T' = < f(0) > \end{pmatrix}. (10)$$

If $dis(q_x, q'_x) \leq n - k$, the polynomial *f*(*x*) can be reconstructed with the assistance of *P* thus the geotag *T* can be reproduced, such that *T*'=*T*. The effective geotag length is *k*log*N*.

Combination Use of Fuzzy Extractors

We design the Euclidean metric fuzzy extractor to adjust the errors introduced by random noises and seasonal biases. The RS and secret sharing based fuzzy extractors can be used to reproduce geotags while location parameters are missing due to offline transmitters.

As noises and biases are always presented in RF signals, Euclidean fuzzy extractor should be applied all the time to minimize the impact of signal temporal variations and guarantee the reproducibility of geotags. Unlike noises and biases, errors due to missing parameters are infrequent. Users have their choices to use which fuzzy extractor. A combination use of Euclidean metric and Hamming metric fuzzy extractors can achieve more robustness in tags but the tradeoff is more entropy loss.

Reproducibility Analysis

In this section we examine and compare the performance of three fuzzy extractor constructions. The evaluation is based on the user's FRR, the attacker's successful rate FAR, and the entropy loss.

All the three constructions improve the consistency of location parameters, thus reducing the geotag FRR. Users' false reject depends on the variations of the parameters, the selected quantization step Δ, and the quantization offset that is, how far off are the received parameters from the center of the quantization grid. The most desired scenario is the distribution of the parameter is exactly in the middle of the quantization grid (offset = 0) whereas the worst case is that the distribution lies on the boundary of the grid (offset = 0.5Δ), shown in Figure 7.

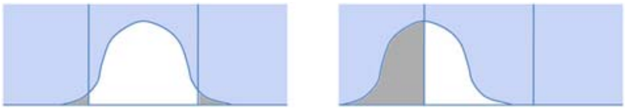


Figure 7. Quantization scenarios: best (left); worst (right)

a) Euclidean Metric Fuzzy Extractor

We first examine how the reproducibility of geotags improves using the Euclidean metric fuzzy extractor. The analysis is illustrated in Figure 8. The x-axis is the quantization steps in terms of standard deviation σ and the y-axis is the estimated FRR. The tag is computed from the triple (TD, ECD, SNR) using the seasonal data from four west coast stations. As a result, there are 11 different location parameters. To estimate FRR we take the first day of the 90-day data as calibration to compute a geotag and the data from the rest of 89 days for verification. The experimental FRR is the number of data points, in which the geotags are matched with the computed tag on day one, divided by all the data points in 89 days. We observe that the estimated FRR is reduced by 84% after applying the Euclidean metric fuzzy extractor.

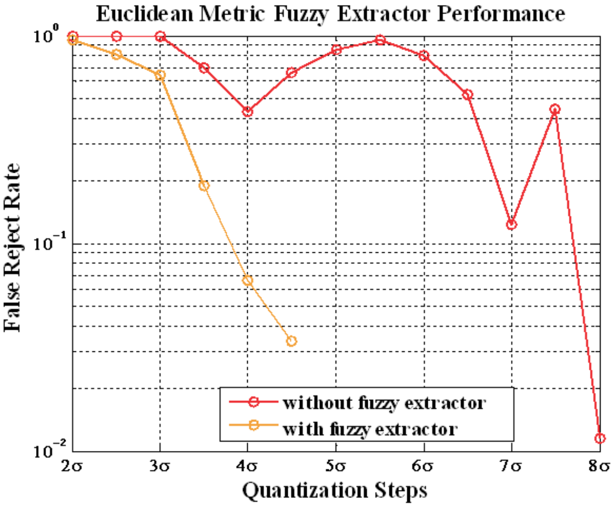


Figure 8. Euclidean metric FE performance improvement

From the mathematical analysis the Euclidean metric fuzzy extractor rounds off the location measurements at verification step to the measurements at calibration step. A geotag can be reproduced when the offset between the two measurements is less than a threshold, Δ/2.

b) RS-Based Hamming Metric Fuzzy Extractor

In practice, multiple parameters are used for the robustness and security strength of geotags. More location parameters provide more information entropy, better resolution, and increase the difficulty in predicting a geotag. However, one drawback is that the FRR of the system is increased. The reproducibility comparison with and without a Hamming metric fuzzy extractor is illustrated in Figure 9. Both cases use Euclidean metric fuzzy extractor to ensure data lying in the middle of the quantization grids. We use 15 parameters to compute a geotag and estimate FRR in this analysis thus *n* = 15. The overall FRR of Euclidean metric fuzzy extractor can be estimated as $1\text{-}\prod_{i=1}^{n}(1-p_i)$, where p_i is the error rate of one parameter or symbol error. We choose the number of errors *t* can be corrected in Hamming metric fuzzy extractor as 2. This results in that *k* = 11. The solid lines represent the analytical analysis while the dots are estimated using the same seasonal data mentioned in the previous section.

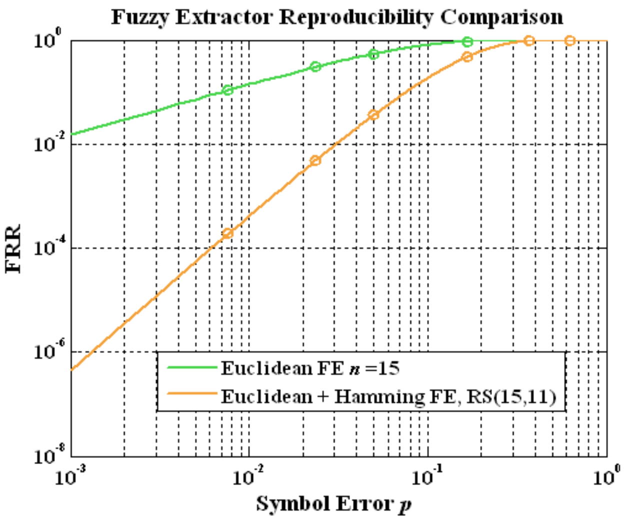


Figure 9. Performance of RS-based fuzzy extractor

PATTERN CLASSIFICATION

To improve the spatial decorrelation geotags and minimize the false reject rate introduced by quantization, we develop a new geotag generation algorithm using pattern classification.

Pattern classification [13] is the concept of assigning a physical object or measured data to one of the pre-specified groups, called *classes*, using *a priori* knowledge or statistical information. The *patterns* are the evaluated final decision from *classifiers* and represent the characteristics of features. Mathematical models are used as the theoretical basis for the classifier design. In classification, a pattern is referred to as a pair of variables {*x*, ω}, where *x* is a collection of features and ω is the concept associated with the features, also called *class label*.

The quality of features is related to the ability to discriminate measurements from different classes. Our goal is to maximize the differences between classes and minimize the inter-class scatter with the extracted decision rules from measurements, thus assigning class labels to future data samples.

Various classes of classification algorithms have been developed and successfully applied to a broad range of real-world domains. It is essential to ensure that the classification algorithm matches the properties of collected data, and to meet the needs of the particular applications. In this paper we select three classifiers—linear discriminant analysis (LDA), k-nearest neighbor (kNN) and support vector machines (SVM)—to implement and generate a geotag.

Linear Discriminant Analysis

LDA is a traditional feature extraction method that aims for a transformation matrix that provides the optimal separation of multiple classes [13]. Data of all different classes are projected onto a subspace in which the data of different classes are as far apart as possible, whereas the data of the same classes are as close as possible. The optimal projection can be obtained by simultaneously minimizing the within-class scatter matrix norm and maximizing the between-class scatter matrix norm.

Fisher's linear discriminant is the classical example of the linear classifier for two classes [14]. The between-class and within-class scatter matrices S_B and S_w are defined by

$$S_B = \frac{1}{M}\sum_{i=1}^{c} l_i(\mu_i - \mu_0)(\mu_i - \mu_0)^T , \tag{11}$$

$$S_W = \frac{1}{M}\sum_{i=1}^{c}\sum_{j=1}^{l_i}(x_{ij} - \mu_i)(x_{ij} - \mu_i)^T , \tag{12}$$

where x_{ij} indicates the *j*th training sample in class *i*, *c* is the number of classes, l_i denotes the number of training samples in class *i*, *M* is the total number of training samples, μ_i is the mean of the training samples in class *i*, and S_W denotes the covariance matrix of samples in class *i*.

The generalized Fisher criterion is defined by

$$J(W) = \frac{W^T S_B W}{W^T S_W W} , \tag{13}$$

where *w* is the generalized eigenvectors of $S_B W = \lambda S_W W$ corresponding to *d* largest eigenvalues.

k-Nearest Neighbor

The kNN classifier is a method for classifying data based on the distance or closeness to the training samples in the *feature space*. A similar idea for geotag generation was proposed in our previous study under the name nearest neighbor method (NNM) in [15].

The method relies on training samples about matching probabilities to consider the *k*-nearest neighbor rule [13]. The class labels are random variables and independent from each other; each has the probability of $P(\omega_i|x)$. The kNN rule selects ω_m with probability $P(\omega_m|x)$ if a majority of the *k* nearest neighbors have a label of ω_m. The value *k* is a design parameter, that is, the probability to select ω_m is larger if the value of *k* is greater. Large *k* reduces the impact of noise and produces smoother decision boundaries, but requires higher computation power. When *k*=1, kNN becomes the nearest neighbor method.

Support Vector Machines

SVM aims to minimize the structural risks. It not only classifies all the training samples correctly, but maximizes the margins between different classes. The problem of overfitting, which degrades the generalization ability, might occur while maximizing the classification performance. In our problem, high generalization ability results in a low FRR. By controlling model complexity, the simplest model that explains data is preferred to avoid overfitting [16].

Let M n-dimensional training samples x belong to two classes. With linearly separable data, the decision function, also referred to as the hyperplane, can be defined as

$$g(x) = w^T x + w_0, \tag{14}$$

where w is an n-dimensional vector and w_0 is a bias term. The problem of deciding the optimal separating hyperplane can be formulated as

$$\text{minimize } J(w) = \frac{1}{2}\|w\|^2, \tag{15}$$
$$\text{subject to } y_i(w^T x_i + w_0) \geq 1, i = 1,2,\ldots, M .$$

If the training data are not linearly separable, the computed classifier may not have high generalization ability even with optimal separating hyperplanes. As a result, to enhance linear separability, the original data are mapped into a higher dimensional space in which data are more linearly separable.

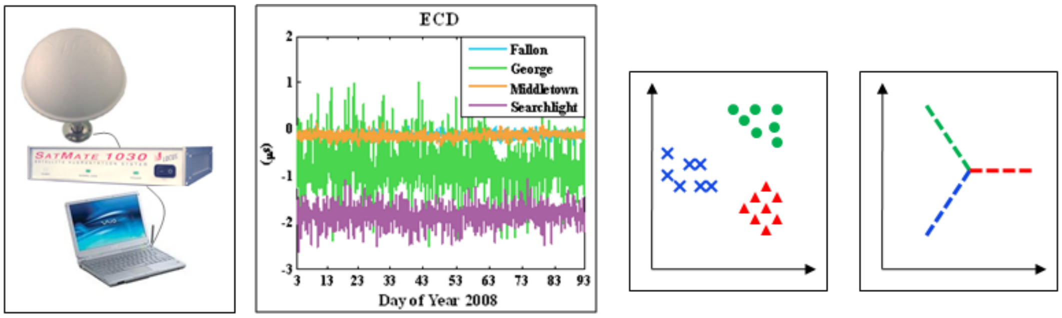
While the SVM classifier maximizes the generalization ability, it is vulnerable to outliers due to the use of sum-of-square errors. Outliers should be mitigated before training to prevent their effects. A margin parameter C controls misclassification errors. A large value of C results in small hyperplane margin and good generalization ability, thus suppressing misclassification errors, whereas a small value of C results in large hyperplane margin and more misclassification errors.

Classifier-based Geotag Generation

To develop an effective geo-security system using pattern classification, it is essential to acquire a thorough understanding of the input feature space and develop proper mapping of such feature space onto the output *classification space*. The machine learning approach we proposed adopts representative statistical models to extract the characteristics of patterns in the feature domain. Different machine learning models should be selected based on the perspective of applications. Practically, the machine learning models have been adopted to construct a robust information processing system for other authentication systems, such as biometrics. The technique is potentially useful in a broad spectrum of application domains, including but not limited to biometrics and geo-security.

The dimension of data is the number of random variables that are measured on each observation. A higher dimension of data, or more features to compute a geotag, results in high spatial discrimination in a geo-security system, as well as total information entropy in a geotag. In practice, however, the added features may actually degrade the geotag reproducibility or reliability of the system, which significantly depends on the training sample size, the number of features for geotag generation, and the algorithm complexity. Such a phenomenon is referred to as the "curse of dimensionality." Dimensionality reduction, which constructs a low-dimensional representation of high-dimensional data, is a means to avoid the curse of dimensionality and improve computational efficiency, classification performance, and the ease of modeling.

Figure 10 illustrates how to generate a robust geotag using pattern classification. The system also works in two steps: calibration and verification. Both steps involve data collection, signal processing, feature extraction,



Data collection

Signal processing

Dimensionality reduction

Classification

Figure 10. Pattern classification-based geotag generation

dimensionality reduction, and classification. At calibration, a model is determined based on the training data. The model should be saved for future classification at the verification step. The geotag T_i associated with location i is obtained from the class label ω_i, such that $T_i = f(\omega_i)$, where $f(\cdot)$ is a mapping function. An example of a mapping function can be a hash function, which is a fundamental block of many cryptographic algorithms. All of the computed geotags will be stored on the database. At verification, the developed model is applied to classify the reduced dimension data; a new geotag is computed using the same mapping function from the extracted class labels. The matching algorithm to decide whether the computed geotag is authentic or not, is the same as the one for the quantization-based geotag matching.

Experimental Results

This section evaluates LDA, kNN, and SVM-based geotags in terms of spatial discrimination and geotag reproducibility using multiple Loran data sets. Spatial discrimination or decorrelation is significant to the security level of a geo-security system. A geotag with high spatial decorrelation ensures that users at different locations with small separation can achieve different geotags, thus lowering FARs. The system reliability depends on geotag reproducibility, and is quantified using FRR.

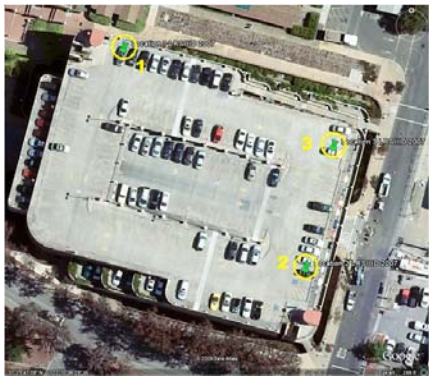


Figure 11. Test locations in a parking structure at Stanford University

The first data set was collected at three test points in a parking structure at Stanford University to examine the three classifiers. A visualization of the three locations in green markers is shown in Figure 11.

The same features – TD, ECD, and SNR – from four West Coast stations are used to derive a geotag. As a result, the input location feature vector is 11-dimensional. A linear dimensionality reduction algorithm is applied to lower the input vector dimension to two to achieve better spatial decorrelation.

a) LDA

The two-dimensional data $[x^1, x^2, x^3]$ that represent three locations are labeled classes 1, 2, and 3 and plotted in Figure 12. The estimated classifier is visualized as a separating surface, which is piecewise linear. The input data were trained using the Perceptron learning algorithm, which minimizes the distance of misclassified points to the decision boundary. The algorithm is an iterative procedure that builds a series of vectors $[w;\ w_0]$ until the inequality condition is satisfied. The inequality is represented as

$$[w; w_0]\cdot z_i^{\,y} > 0, \quad i = 1,\ldots,3,$$
$$z_i^{\,y}(j) = \begin{cases} [x_i^{\,T},1]^T, & \text{for } j = y^i, \\ -[x_i^{\,T},1]^T, & \text{for } j = y^i, \\ 0, & \text{otherwise.} \end{cases} \qquad (16)$$

There is more than one solution when the input data are separable. The final solution depends on the initial vector $[w;\ w_0]^{(0)}$, which can be selected arbitrarily. The algorithm does not converge when the data are not separable.

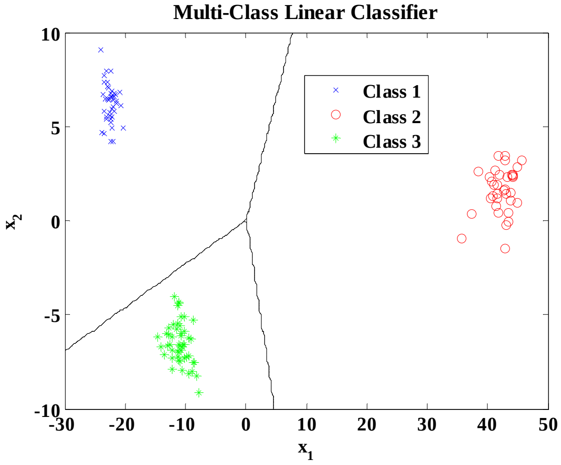


Figure 12. Multi-class linear classifier trained by the Perceptron algorithm

b) kNN

The best choice of k depends on the input data; large values of k reduce the effect of the noise. The decision boundaries of the case k=8 are plotted in Figure 13. The algorithm is easy to implement but computationally intensive, especially when the training data size grows. Euclidean distance is used to measure the closeness between samples.

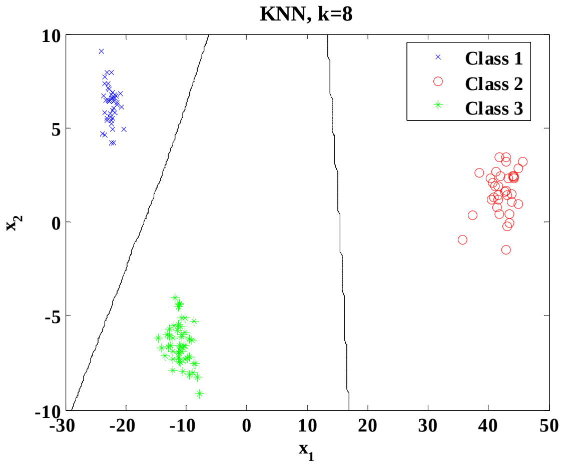


Figure 13. The decision boundaries of kNN, k=8

c) SVM

As mentioned earlier, SVM is considered as an optimization problem. To solve the optimization, Sequential Minimal Optimization (SMO) is applied. The One-Against-One (OAO) decomposition is used to train the SVM classifier. An input parameter, kernel argument, controls the size of the hyperplane margin, thus adjusting the misclassification errors.

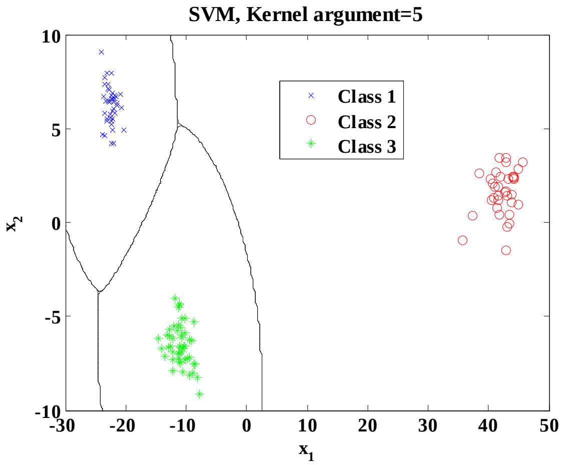


Figure 14. Multi-class SVM classifier by OAO decomposition

d) Spatial Discrimination

Another data set was collected in the same parking structure to evaluate and compare the spatial decorrelation of different classifiers. Eleven test locations aligned in a straight line were chosen with a separation of three meters. The same 11 location-dependent parameters are the inputs to the geotag generation algorithm.

The performance metric for spatial decorrelation is FAR. The first test point was selected as a master location, or an authentic user, while the rest of the test points are seen as attackers. Three different geotag generation algorithms – SVM classifier-based, kNN classifier-based, and quantization-based – are compared; the estimated FARs are illustrated in Figure 15. The result indicates that the kNN classifier-based geotag has the best discrimination or spatial decorrelation, whereas the quantization-based geotag has the worst, since the error rate decreases slowly as the attacker moves away from the master location.

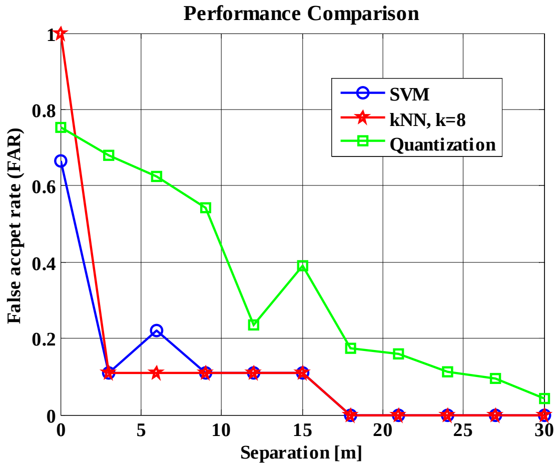


Figure 15. FAR comparison of different geotag generation algorithms

CONCLUSION

We proposed location-based security services using RF signals in which location is used as a validation to restrict or deny certain actions in security applications. Geotags are computed from location information that is obtained from a location sensor. The geotag is not a replacement but builds on the conventional authentication schemes. This location-based service can be applied to many applications, such as Loopt, DMP, inventory control, and data access control.

We developed fuzzy extractors, which are the error tolerant algorithms to recover secret information from noisy location data. A Euclidean metric fuzzy extractor was proposed to deal with noise, biases and quantization errors to achieve high reproducibility of geotags. The Reed-Solomon based and secret sharing based fuzzy extractors were designed for the scenario in which RF transmitters are offline. One drawback of Hamming metric fuzzy extractors is the entropy loss in the computed geotags.

Classifier-based geotag generation algorithms were proposed to achieve high spatial discrimination. The pattern classification uses machine learning techniques that improve not only the spatial decorrelation of computed geotags but also users' convenience. The location data can be trained automatically based on the classifiers. The three classifiers proposed on location data are LDA, kNN, and SVM.

Real location data were used to evaluate the performance of the classifier-based geotag generation methods in terms of FAR and FRR. According to the comparison result, both kNN and SVM classifier-based methods can result in good spatial discrimination. Future study includes investigating other effective classifiers to improve the performance of geotag generation methods.

ACKNOWLEDGMENTS

The authors would like to thank Mitch Narins of the FAA Loran Program Office for supporting this effort. In addition, we would like to thank Dr. Greg Johnson, Ruslan Shalaev, and Christian Oates from Alion Science & Technology for providing the Stanford Seasonal Monitor data collection equipment. Finally, thanks to the US Coast Guard (USCG) Loran Support Unit (LSU) for the use of their equipment in our testing.

REFERENCES

[1] J. Hruska. “Microsoft patent brings Miss Manners into the digital age.” June 11, 2008.

[2] B. Schneier. “Kill switches and remote control.” A blog covering security and security technology. July 1, 2008.

[3] L. Scott and D. Denning. “Location based encryption & its role in digital cinema distribution.” *Proceedings of ION GPS/GNSS 2003*, pp288-297.

[4] D. Qiu, S. Lo, and P. Enge. “Geoencryption using Loran.” *Proceeding of ION NTM 2007*.

[5] D. Qiu. “Security analysis of geoencryption: A case study using Loran,” *Proceeding of ION GNSS 2007*.

[6] S. Lo, R. Wenzel, G. Johnson, and P. Enge. “Assessment of the methodology for bounding Loran temporal ASF for aviation.” *Proceeding of ION NTM 2008*.

[7] P. Swaszek, G. Johnson, R. Hartnett, and S. Lo. “An investigation into the temporal correlation at the ASF monitor sites.” *Proceedings of ILA 36th Annual Meeting 2007*.

[8] D. Qiu, S. Lo, and P. Enge. “A measure of Loran location information.” *Proceeding of ION PLANS 2008*.

[9] A. Juels and M. Wattenberg. “A Fuzzy Commitment Scheme”. *Proceedings of ACM Conf. on Computer and Communications Security*, pp28-36, 1999.

[10] Y. Dodis, L. Reyzin, and A. Smith. “Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data”. *Eurocrpt’04*, vol. 3027 of LNCS, pp523-540. Springer-Verlag, 2004.

[11] T. Cover, Elements of Information Theory. John Wiley & Sons, Inc. 2001.

[12] A. Daraiseh and C. Baum. “Decoder Error and Failure Probabilities for Reed-Solomon Codes: Decodable Vectors Methods”. *IEEE Trans. Commun. 46(7), July 1998, pp. 857-859*.

[13] R. O. Duda, P. E. Hart, and D. G. Stork (2001) *Pattern classification* (2nd edition), Wiley, New York.

[14] X. Wu, K Wang, and D. Zhang. “Palmprint recognition using fisher's linear discriminant.” *Proceedings of International Conference on Machine Learning and Cybernetics 2003*.

[15] D. Qiu, D. De Lorenzo, S. Lo and P. Enge, “Physical pseudo random function in radio frequency sources for security.” *Proceeding of ION ITM 2009*.

[16] S. Abe. Support vector machines for pattern classification. Springer-Verlag New York, 2005.