Geoencryption Using Loran

Di Qiu, Sherman Lo, Per Enge, Dan Boneh, *Stanford University*
Ben Peterson, *Peterson Integrated Geopositioning*

BIOGRAPHY

Di Qiu is a Ph.D. candidate in Aeronautics and Astronautics working in the Global Positioning System (GPS) Laboratory at Stanford University. Her research focuses are geoencryption and signal authentication.

Dr. Sherman Lo is currently a research associate at 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.

Dr. Benjamin Peterson spent most of his career on the faculty of the U.S. Coast Guard Academy. He retired from the Academy in 2000. Afterwards, he founded Peterson Integrated Geopositioning.

ABSTRACT

Geoencryption is the use of position navigation and time (PNT) information as means to enhance the security of a traditional cryptographic system. The information is used to generate an additional security key, a "geolock", that is necessary to access the encrypted data or application. The concept was originally proposed by Logan Scott and Dorothy Denning. This paper examines the benefits of using Loran for geoencryption and the implementation of geoencryption on Loran.

INTRODUCTTION

Traditional encryption is used to provide assurance that only authorized users can use the secure content. However, it would still be useful to have an additional layer of security that provides assurance that the secure content can only be used at authorized location and time. The concept of location based encryption or geoencryption is being developed for such a purpose. The capability has tremendous potential benefits to applications such as managing classified/secure data and digital movie distribution where controlling access is the predominate concern [1].

To implement geoencryption, in principle, a device performing the decryption integrates a location sensor and cryptographic algorithms. Different radio frequency (RF) signals were studied and compared. Loran is chosen as a case study due to its many properties that are useful to geoencryption. A practical concern for implementing this device is whether it can be made resistant to unauthorized used and "tampering". By tampering, we mean both physical attacks on the hardware and attacks on the implementation such as spoofing. If the device is vulnerable to tampering, it may be possible to for an adversary to modify it and bypass the location check [2]. To protect against tampering and spoofing, a signal authentication protocol, Timed Efficient Stream Loss-tolerant Authentication (TESLA) is proposed. We propose a mean on implementing TESLA on Loran for authentication.

The structure of this paper is as follows. The paper first describes how the geoencryption builds on conventional cryptographic algorithms and protocols and provide an additional layer of security. The paper then discusses the properties of Loran, which are robust for geoencryption approach. The paper then provides a detailed discussion of TESLA and its implementation on Loran. Stanford University is developing a geoencryption testbed that uses Loran as an input to investigate the feasibility of the algorithm. The paper concludes with some preliminary results from the testbed.

GEOENCRYPTION

Before discussing geoencryption and its implementation, a review of some cryptographic terms, concepts and algorithms will prove useful.

Review on Cryptographic Concepts

The basic goal of most cryptographic system is to transmit some data, termed the plaintext, in such a way that it cannot be decoded by unauthorized agents. This is done by using a cryptographic key and algorithm to convert the plaintext into encrypted data or ciphertext. Only authorized agents should be able to convert the ciphertext back to the plaintext.

A cryptographic algorithm, also called cipher, is used to perform the transformation. The cipher is a mathematical function that used for encryption and decryption. There are two general types of key-based algorithms: symmetric and asymmetric (or public-key). Symmetric algorithms are the algorithms where encryption key can be calculated from decryption key and vice versa. In most symmetric algorithms, the encryption key and the decryption key are the same as shown in Figure 1. These keys are often called session keys. Public-key algorithms are designed so that the keys used for encryption and decryption are different as shown in Figure 2. These keys cannot be mutually derived – i.e. you cannot derive the decryption key from the encryption key. The encryption key is often called the public key and the decryption key is called the private key [3].

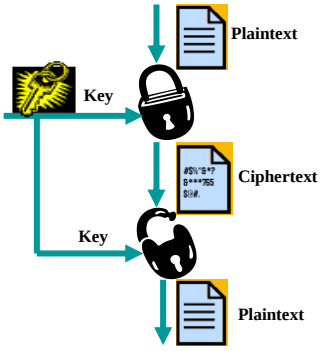

Figure 1: Symmetric Algorithm

The most widely used symmetric algorithms are DES, Triple-DES and AES. The most popular public-key algorithm in use today is RSA, developed by Rivest, Shamir and Adleman [3]. AES and RSA will be used to implement our demonstration geoencryption protocol.

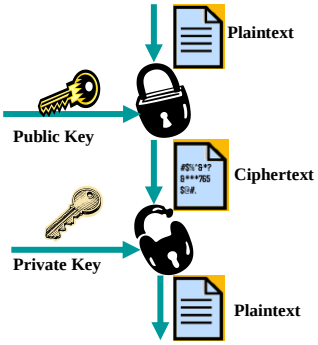

Figure 2: Public-key/Asymmetric Algorithm

There are two reasons why public-key algorithms are not used interchangeably with symmetric algorithm. First, public-key algorithms are slow, about 1000 times slower than the symmetric algorithms. Second, the public-key cryptosystems are vulnerable to chosen-plaintext attacks. Therefore, in most practical implementations, public-key algorithm is used for key management, to secure and distribute session keys. The plaintext is encrypted using symmetric algorithm. This is called a hybrid algorithm [3].

Authentication is another important concept in cryptography. It allows the receiver of a message to ascertain its origin. Authentication is not necessarily used in encryption or decryption protocols but it is a key concept in verifying the source of a message. It will be used for signal authentication which will be discussed in the later section. Hash functions are a fundamental building blocking for many of the authentication protocols. A hash function is a function that takes a variable length input and converts to a fixed length output, called hash value or hash digest [3]. Hash functions are relatively easy to compute but significantly harder to reverse. Beside one-way-ness, the other important property of hash functions is collision-free: It is hard to generate two inputs with the same hash value.

A message authentication code (MAC), also known as data authentication code (DAC), is a one-way hash function with the addition of a key. The hash value is function of both of the input and the key [3]. Unlike encryption, authentication doesn't hide the plaintext but tag the MAC at the end of the plaintext for the recipient to verify whether the plaintext has been modified on the way of distribution.

The Geoencryption Example: Digital Film Distribution

A particular application of geoencryption is for digital film distribution. The idea of geoencryption and its use in digital film distribution was proposed and developed by Logan Scott, Dr. Dorothy Denning, and their colleagues at Geocodex [1]. The overview of the system is shown in Figure 3. Under this system a content provider ("sender") distributes the encrypted film (ciphertext) to an authorized user ("recipient"). This is done via many methods (such as satellite data links) and, as such, may be readily available to unauthorized users. The desire is to have films encrypted using the geoencryption protocol that is decryptable only at a specified location (theaters). The desire is for the decryption process to fail and not reveal information about the plaintext should there be an attempt to decrypt the data at another location, This should be true whether it is by an authorized or unauthorized user, Therefore, the geoencryption algorithm can be used to

ensure that film cannot be retrieved except at the theater by authorized personnel.

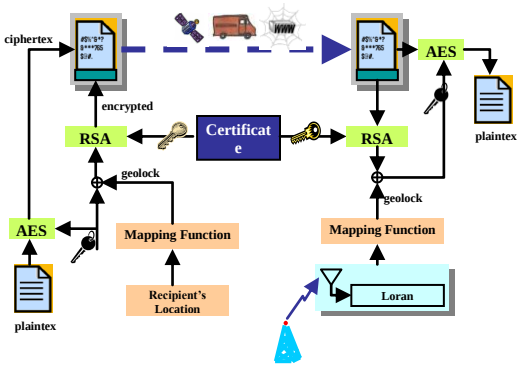

Figure 3 : Geoencryption Overview

Traditional encryption is an integral part of the system. The sender encrypts the data file or plaintext using AES, a symmetric cipher, using a random key. The random key is transmitted to the authorized user after being encrypted with a key ("geolock") derived from specific user location (and time) dependent parameters. The geolock is generated by mapping the recipient's location information into binary bits. And the geolock encrypted key is then encrypted again with a public-key cipher, such as RSA. To ensure authenticity of the sender/receiver, both the public key and the private key are distributed by a trusted third party, Certification Authority (CA).

In order to enable the geoencryption system, the recipient should have three channels to receive information. First, a data receiver is needed to capture of digital encrypted data file. Furthermore, a navigation receiver is needed to receive RF signals whose location dependent parameters are needed to generate the geolock. A third channel is necessary for secure key exchange. If geolock is correct, the decryption process is performed using the right random key and received encrypted data file.

Location Entropy

Location itself may not be adequate for generating the geolock due to the insufficient randomness or entropy. Suppose we divide the entire earth into small grids and uniquely represent each grid box. The grid size depends on the accuracy of our positioning sensor. The information content available is the minimum of number of bits necessary to do that representation. With a very high performance hardware brute-force attack, time required to finish searching all the possible combinations can be estimated. Table 1 shows the equivalent number of bits and time to break it using a $10,000 hardware attack in 2005. The brute-force attack machine is built with key search chips. The chip can test keys at a rate of 50 million per second in 1995. The more chips used, the less time and more cost to search an entire key space. The above cost takes into account the device cost goes down a factor of 10 every five years [3].

Grid Space (m)	# of Bits	Brute Force Attack Time
10-4	75	283 years
10-3	69	2.8 years
10-2	62	10 days
10-1	56	2.5 hours
1	49	1.5 minutes
10	42	< 1 second
102	36	< 1 second
103	29	< 1 second

Table 1: Grid spacing accuracy versus data required for representation

LORAN FOR GEOENCRYTPION

Signal Requirements for Geoencryption

With an understanding the objectives and approaches of geoencryption, we now can examine RF signals and their properties. In particular, we want to identify location dependent signal characteristics that adapt well for use for geoencryption.

First, the signal parameters should be location dependent only and minimally sensitive to temporal. This implies the repeatability and repeatable accuracy is important. This allows a recipient to provide his location-dependent parameters or the derived geolock to the sender at one time and still have those parameters valid at a latter time. In other words, the signal characteristics should be consistent enough that when the recipient is ready to decrypt, measurements at the same location will yield the same geolock that was previously generated.

Second, there should be adequate location dependent information to generate a reasonably strong geolock key.

Third, it is capable of anti-spoofing. If the signal is vulnerable to spoofing, it may be possible for an adversary to bypass the location check and decrypt correctly.

Furthermore, it is desirable that the signal is available indoors. This is desirable as many of the anticipated application of geoencryption will likely occur indoors.

This includes applications such as the management and distribution secure digital data. Often, it is desired that this data is only accessible inside certain building(s).

Loran's Potentials for Geoencryption

Loran is a terrestrial, low frequency pulsed navigation system that operates in much of the northern hemisphere, has many properties that are useful to geoencryption. Furthermore, it is being modernized to a next generation system known as enhanced Loran (eLoran) [4] which will have additional capabilities that can benefit its use for geoencryption.

First, Loran has good repeatable accuracy in position, which benefits the design of the geolock. Figure 4 shows position scattered plot. The data was collected in Stanford University for several hours on Jan 8th, 2006. The position error in east-west direction is less than 10 meters and the error in north-south direction is less 25 meters.

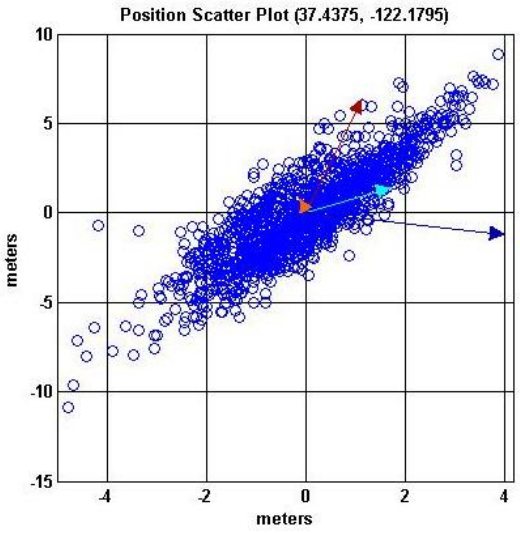

Figure 4: Position Scatter Plot from Loran as measured at Stanford University

Second, Loran is transmitted from static transmitters and, as a result, there are many parameters that are location dependent. This is important as the security strength of the geolock is derived from the information content or entropy of the information used to generate it. More parameters as well as increased accuracy of those parameters, increases the entropy. Signals from static transmitters may have many location dependent characteristics or parameters. The possible useable Loran parameters are time difference of arrival (TDOA), envelope to cycle difference (ECD), difference of signal to noise ratio (ΔSNR), and shape of the envelope.

Third, Loran is a high power low frequency signal. This means it is hard to spoof and hard to jam. Furthermore, the signal can reach some places such as urban canyons and indoor environment that may not be reachable by a line of sight system such as GPS.

Finally, Loran has a data channel that can carry authentication and time messages. Both of them are important to the authentication scheme we propose on Loran. Authentication message is used to provide the verification of the source of the Loran signals and time message helps synchronizes the user and the Loran transmitters.

SIGNAL AUTHENTICATION ON LORAN

The purpose of geoencryption is to provide security to the transmission of information. As such, it is important that every linkage of the geoencryption chain is secure. This includes not only the protocol itself but also the broadcast of RF signal. The basic protocol is discussed previously in [5]. The security of the RF navigation signal is provided by message authentication. Authentication is about the verifying the source of the data/messages. One goal is to prevent the user from being fooled into believing that a message comes from a particular source when this is not the case. Another goal is to allow the receivers to verify whether the messages have been modified during transmission.

The main challenge of secure broadcast communication is source authentication, and the problem is complicated by untrusted or uncertified users and unreliable communication environments. The concern is that untrusted users may employ items such as signal simulator to spoof the system into generating the correct geolock. Source authentication helps the receivers to verify the received data originates from the source and has been modified in transit.

Furthermore, adding security in a broadcast communication system is difficult because symmetric authentication algorithms are fast and efficient but not as secure as asymmetric ones in a broadcast setting; on the other hand, the asymmetric authentication algorithms are secure but not efficient. Therefore, we propose TESLA on Loran to provide authentication and improve system integrity. TESLA uses symmetric authentication mechanism by appending MAC at the end of each message, which is transmitted from a sender to a receiver, and time (delayed key disclosure) to achieve asymmetry property required for a secure broadcast authentication [5]. The main features of TESLA are: low sender and receiver computation overhead, low communication overhead, and perfect robustness to message loss. It requires buffering for both sender and receiver sides but the receiver can

authenticate the message as soon as enough messages, keys and MACs are buffered [5].

Loran Data Channel (LDC)

Enhanced Loran will transmit data via a data channel. The current proposal is ninth-pulse modulation [6]. The modulation is chosen to minimize the impacts on the current operational Loran signal. An additional pulse is inserted after the eighth pulse of pulse group of secondary stations, shown in

Figure **5**. Third-two state Pulse Position Modulation (PPM 32) is used to change the time delay of the ninth pulse from 1000 microseconds after the eighth navigation pulse [6].

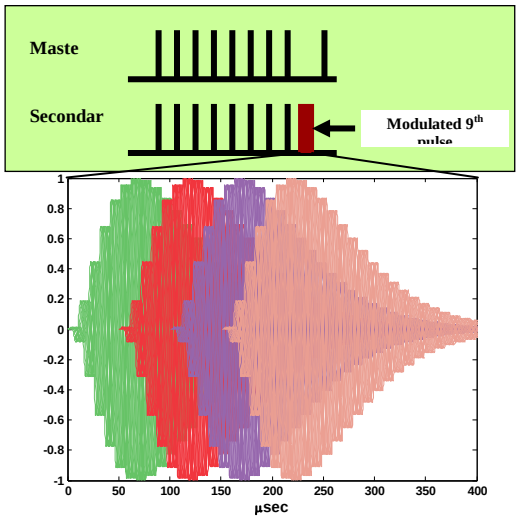

Figure 5: PPM-32 on 9th Pulse

The delays of the 32 symbols are given in the formula from zero-symbol offset:

$$d_i = 1.25 \bmod(i,8) + 50.625\, floor(\frac{i}{8})$$

Under the current proposed ninth pulse communications, each Loran message has 120 bits and consists of a 4-bit header, a 41-bit payload, and 75-bit parity component. Some of the message types have been defined such as differential Loran corrections which provide phase correction at known reference sites, almanac, message for government use and time of the day. There are some types are undefined and reserved for future use. The Reed-Solomon codes are used for parity check. This forward error correction coding method provides error correction capacity and integrity. It provides to ability to align the message and to verify that the message has been validly decoded with high probability [7].

The demodulation can be done using matched filter. A matched filter basically performs convolutions of the time-reversed version of a reference signals with the input signal. The demodulation process is complicated by the presence of noise and interference on the input signal. Multiple matched filters, each referenced to a specific state, are used. The input signal passes through each matched filter, shown in Figure 6. A comparator is used to compare the values after the filters, and the maximum value determines the delay and thus the symbol modulated on the pulse [7]. This matched filter model is the model used for the analysis in the later part of this paper.

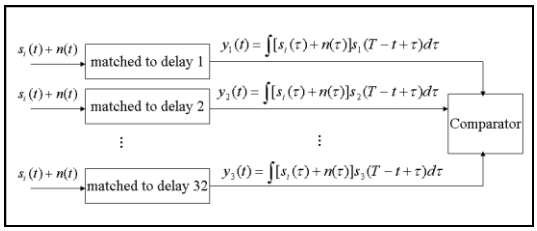

Figure 6: PPM Matched Filter

Implementing TESLA on Loran

First, the Loran transmitter and the receiver should be loosely synchronized in time. The synchronization does not have to be precise but the receiver knows an upper bound on the sender's local time. Therefore, a secure time channel is needed for receivers, either using Internet or Loran time message to achieve this goal. Here is the outline and sketch of the TESLA approach [5]:

- One-way key chain generation: A TESLA chain on size N is selected. The transmitter generates a one-way chain of N self-authenticating values or keys, denoted K_1, ..., K_N, and assigns the keys to the N segments (one segment is the time interval necessary for one authenticaion message) sequentially. A hash function is used to construct the one-way chain and derives from the base key, K_N. The other keys K_i is generated from N-i hashes of K_N. Notationally, $K_i = F(K_{i+1}) = F(F..F(K_N))$ where there are N-i instances of the hash function F. Figure 7 illustrates the construction of one-way key chain and F indicates the hash function used. When the keys are broadcast, the chain is sent in the reverse order of generation.

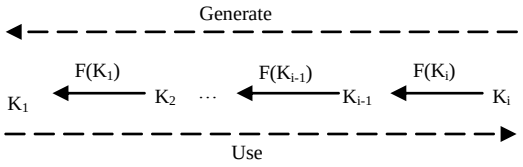

Figure 7: One-way Chain Construction

- MAC key generation: The transmitter uses a different hash function F' to hash the last one-way chain values and results in the keys, K_1', ..., K_N', used to form MACs.

- MAC generation: The transmitter computes the MACs over the contents of the messages and keys and attaches them to each packet. So each segment has the message, the MAC for this message and the key for a previous MAC. And this transmitted keys are the first one-way chain values. An illustration is shown in Figure 8, where the key disclosure delay of one segment is used. For instance, K_i is not disclosed in the segment of M_i but in the segment of M_{i+1}.

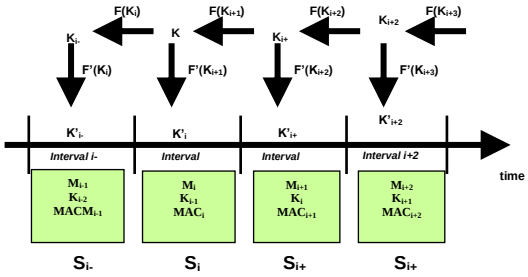

Figure 8: Sender Setup

- Broadcast stage: The messages, keys and MACs are transmitted in segments. Each segment consists of a message, a MAC and a key associated with the message in the previous segment, shown as a green block in Figure 8.

- Key verification: Each receiver buffers the segments first. The first step is to verify the received keys values. This is done by hashing the key in current segment and comparing it with the key in previous segment.

- MAC verification: Each receiver checks the correctness of MAC of buffered segments after the keys are verified. If the MAC is correct, the receiver accepts the segment.

DEMONSTRATION

The west coast chain of Loran, GRI 9940 is used to perform the demonstration. The stations of this chain are Fallon, NV, George, WA, Middletown, CA and Searchlight, NV. Middletown, the closest secondary station to Stanford University, is chosen to implement this authentication scheme to ensure the performance of decoding. Figure 9 shows a picture of the Loran tower at Wildwood, NJ.

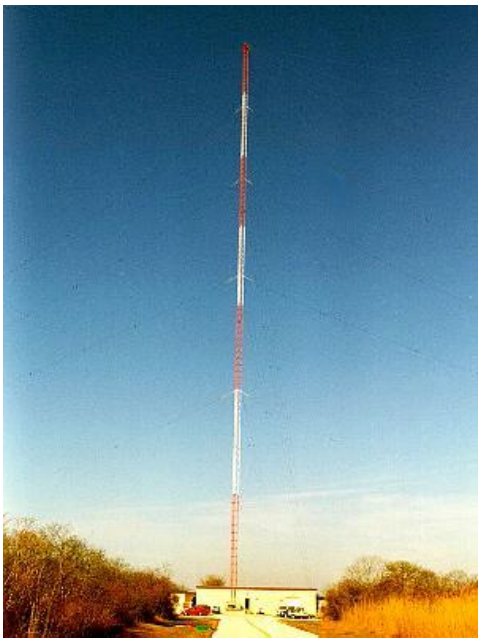

Figure 9: Loran Transmitter at Wildwood, NJ

Preparations on Middletown

Middletown broadcasts both time and authentication messages. The time message is generated by United States Coast Guard (USCG) to test the performance of 9th pulse modulation. Stanford University generates the authentication messages to verify authentication performance and demonstrate geoencryption protocol. The time and authentication messages are broadcasted alternatively. 50% bandwidth is obtained for authentication messages. With only one secondary station is carrying data message, a data rate of 50 bits/sec is achieved.

Two hash functions are necessary to compute the TESLA one-way chain key values. For our demonstration, we chose SHA1 and MD5. SHA1 outputs a hash value of 160-bit and MD5 outputs a hash value of 128-bit. SHA1 is employed in several widely used security algorithms and protocols. While MD5 has been found not to be collision-resistant, it remains the desired property of one-way-ness. Another reason we chose MD5 in this demonstration is its reasonably short digest. HMAC is chosen to generate the MAC and hash function used for HMAC is SHA1, so the MAC size is also 160-bit. The key size to create MAC must be at least half of the MAC

size to ensure the security; hence we choose a key size of 128-bit. The set of MAC keys can be computed using MD5.

Therefore, authentication message consists of key and MAC, and results in a total length 320-bit. With 41-bit payload in Loran messages, at least 8 messages are needed to carry a complete authentication message. Subtypes are used to help the receivers distinguish the MACs and keys in authentication messages. The data type for authentication message is 0011. Subtypes 1 to 4 are for identification of MACs and subtypes 6 to 10 are for keys. Subtype 5 consists of 12-bit MAC, 13-bit padding and 12-bit key. A total of 10 messages are needed to carrier one TESLA packet, and it takes 23.856 seconds to transmit these messages via GRI 9940. The following shows the authentication message structure.

00110001MAC
00110010MAC
00110011MAC
00110100MAC
00110101MAC0000000000000Key
00110110Key
00110111Key
00111000Key
00111001Key

TESLA uses one-way key chain and discloses keys in a delayed manner. The length of the chain depends on the desired time to first authentication and the authentication strength. As such it depends on how much bandwidth is available for authentication. Under TESLA, each segment of the chain consists of a message, a MAC and the delayed key for a previous MAC. The amount of delay is a design parameter. In our proof of concept demonstration, a three segment sequence is used. Additionally, half of the ninth pulse bandwidth is used for authentication messages. The result is that a time message and authentication message are sent alternatively.

In the setup phase, K_1 is generated randomly and the transmitted key chain (K_2, K_3) is computed using SHA1. MD5 is used to generate the keys used for MAC generation. These MAC generation keys are K_1', K_2' and K_3' and they are used with the messages m_1, m_2 and m_3 to compute MAC_1, MAC_2, and MAC_3, respectively. To simply the implementation, three segments are generated and broadcasted repetitively. Figure 10 illustrates the roles of the hash functions and MAC function used and computations of three segments. It is a simplified version of Figure 8.

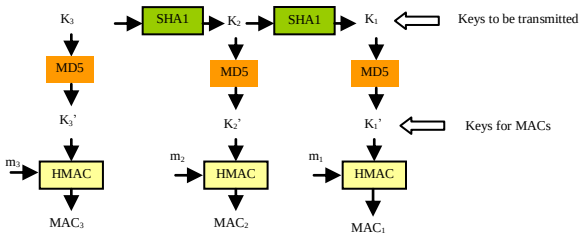

Figure 10: Key and MAC generation for Demonstration

In the broadcast phase, the three segments are transmitted in a sequence of <m_1, MAC_1, K_3>, <m_2, MAC_2, K_1> and <m_3, MAC_3, K_1>. An illustration is shown in Figure 11.

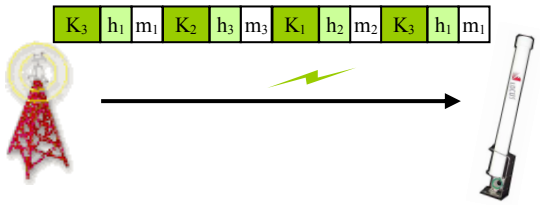

Figure 11: Circular TESLA Chain

K_1, the first key of generation and last key of transmission, is the embedded key in the receiver. Once enough segments are received and buffered. The three steps of verifications are performed.

1. First stage key verification: Compare the received K_1 with the embedded key. If the same, move on to the next step. This verifies that the source is the same as the one that provided the key.

2. Second stage key verification: Hash the received keys using SHA1 and compare them with the keys in the previous packet. This verifies continuity of the source. That is, another signal source has not been injected.

   $SHA1(K_2) ?= K_1$
   $SHA1(K_3) ?= K_2$

3. MAC verification: Construct the MAC keys using MD5 and compute the MACs with these keys and the received messages. Compare these computed MACs with the received ones (h_1, h_2, h_3). The signal is validated if they match. This verifies that the message has not been altered.

   $HMAC(MD5(K_1), m_1) ?= h_1$
   $HMAC(MD5(K_2), m_2) ?= h_2$

$$HMAC(MD5(K_3), m_3) ?= h_3$$

Theoretical Analysis of TESLA Performance

The performance of TESLA depends on the signal to noise ratio (SNR) of the performance of modulation technique and authentication bandwidth. A matched filter model in the presence in noise for the receiver processing of the signal is used to analyze the performance. Additive white Gaussian noise is assumed to pass through the filter. The noise variances are used to determine an upper bound on the probability of error, which is the probability a sent symbol is not correctly received by the receiver, for different SNR [8]. A GRI can carry 5 symbols, which is considered a packet. Once the probability of symbol error is determined, the average and standard deviation of the packet loss rate can be estimated. Assuming that the packet loss distribution is approximately gaussian, the message loss can be calculated using forward error correction (FEC) [8]. Depending on the number of Loran messages needed to carry one TESLA segment, the probability of authentication or probability to verify a TESLA segment correctly can be determined, shown in Figure 12.

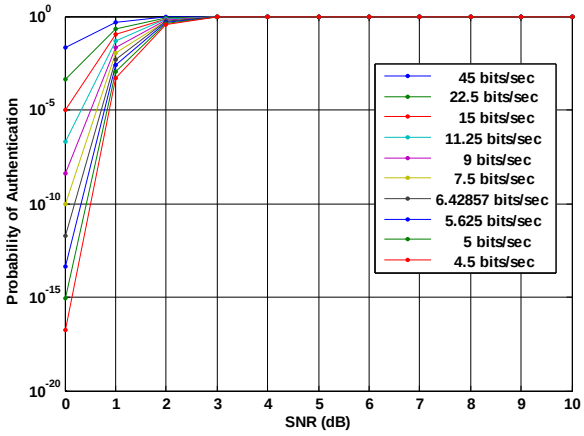

Figure 12: Probability of Authentication

As SNR increases, the probability of symbol error decreases and this results in a decrease of message loss rate and an increase of authentication probability. For each SNR, the probability of message loss is fixed. Depending on the implementation of TESLA, the available bandwidth for authentication messages determines the number of Loran messages required to carry the data messages, keys and MACs. As mentioned in the previous section, each Loran message consists of 120-bit symbols with a payload of 41 bits. Therefore, an increase of authentication bandwidth results in a decrease of the number of Loran messages to carry each TESLA segment. With assumption that each Loran message is broadcasted independently from each other, the probability of authentication can be estimated and it also increases as bandwidth increases.

Another important parameter to test the performance of TESLA is the authentication time, or time of alert. That is the average time that a user is needed before he can authenticate. Similar to the probability analysis, Figure 13 shows the authentication time also depends on SNR and the bandwidth of the authentication message.

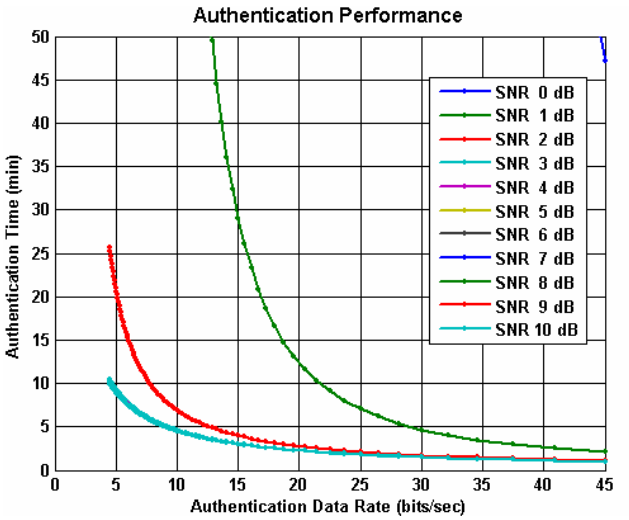

Figure 13: Mean Authentication Time

Receiver HW & SW Development

To test the performance of TESLA and geoencryption, a demonstration testbed is built. The testbed is developed in both hardware and software. Figure 14 illustrates the overall architecture of the receiver

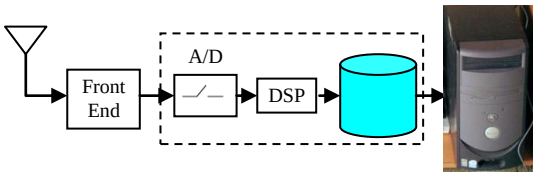

Figure 14: Receiver Architecture

To capture RF signals, a Locus antenna was used. The output of the antenna was connected to a Locus LRS IIID receiver. The receiver was only used to function as a front-end to amplify and filter the incoming RF signals. The output of the Locus receiver goes into ELRR (Enhanced Loran Research Receiver) to first digitize and process the conditioned signals. ELRR also decodes the messages modulated on the 9th pulse of Middletown. A serial port is used to allow MATLAB to communicate with ELRR. The picture in Figure 15 shows the Locus antenna, locus receiver (upper) and ELRR (lower). The

software development of the receiver consists of TESLA authentication using the decoded messages from ELRR, position and location dependent parameters estimation, geolock generation.

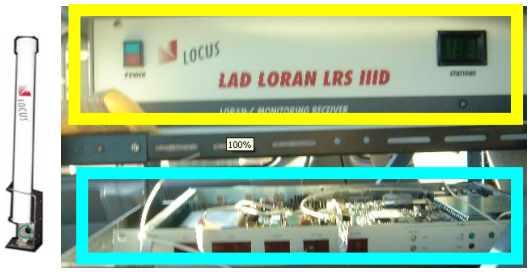

Figure 15: Receiver Hardware Development

To investigate the geoencryption protocol, a MATLAB GUI is built to provide a better visualization, shown in Figure 16. The GUI simulates symmetric cipher, AES, for plaintext encryption, asymmetric cipher, RSA, for random key encryption, and a mapping function for geolock generation. On the receiver side, the hardware setup in Figure 15 is used.

This demonstration simulates the entire geoencryption protocol. The sender is first required to input the plaintext or filename of the plaintext in the editable text box underneath the plaintext icon. Moreover, the location information of the recipient is needed. For simplification, the parameters used to compute the geolock are latitude in degrees, longitude in degrees, time, space grid size in meters and time grid size in hours.

The program also simulates the recipient. The user has an option to choose either using simulated signal or real signal from Loran transmitters. The demonstration assumes ciphertext, public key and private key distributions are secure. The details of the protocol simulations are described as follows:

- Encryption on the sender (content provider) side: The sender first generates a random key. He takes the plaintext and the random key as inputs to AES to generate ciphertext. The location information goes into the mapping function and a geolock is computed. Next, The random key is first XORed with the geolock and this geolocked-key is encrypted using RSA

- Decryption on the receiver (movie theater) side: The user uses Loran antenna and receiver to capture Loran signals and decode Loran messages. Perform TESLA authentication using the decoded messages. Once the authentication is verified, the receiver estimate user location

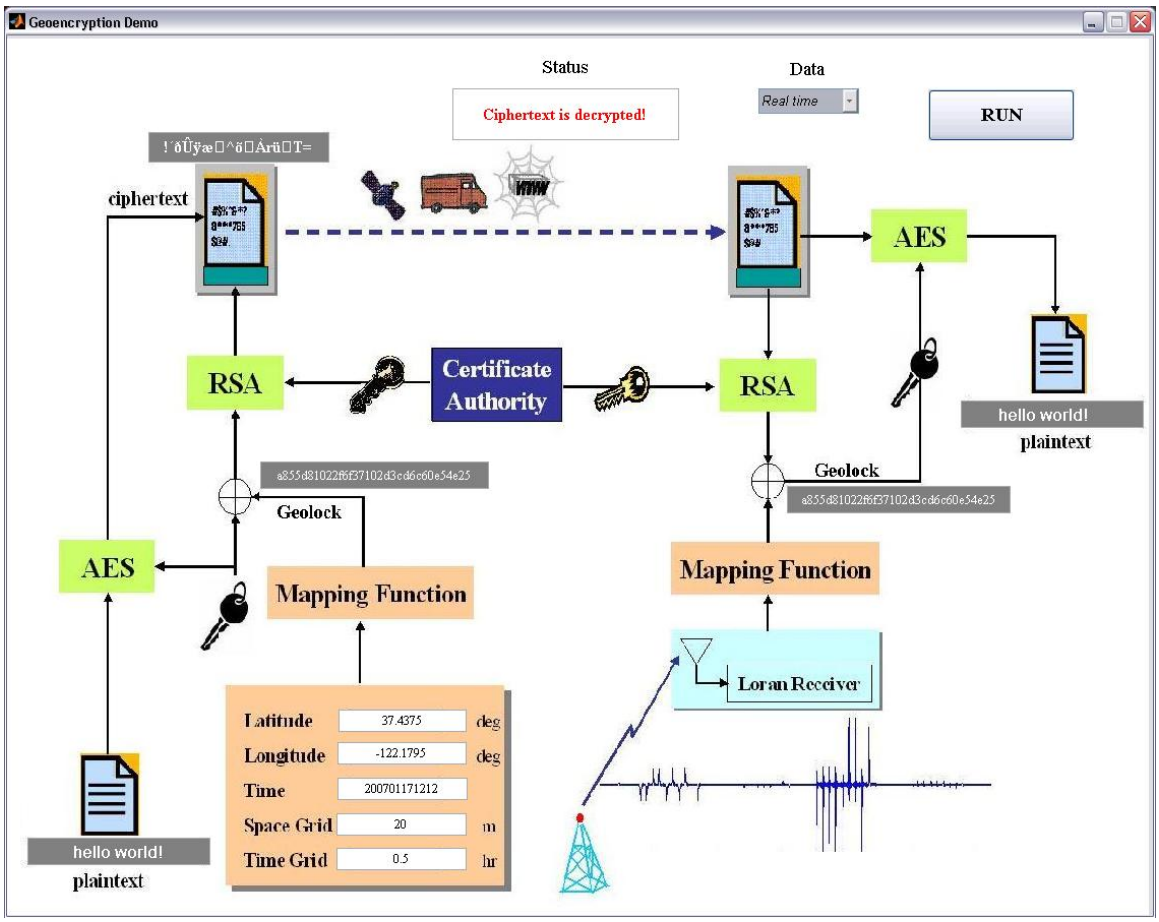

Figure 16: Demonstration GUI. Receiver Location: Palo Alto. Grid Size: 20 meters.

and location dependent parameters and map them to compute a geolock. Next, decrypt the encrypted key using RSA and private key. XOR this with the geolock to get random key. At last, this decrypted random key is used to decrypt the ciphertext in AES.

Preliminary Results

This section describes some preliminary results of geoencryption and TESLA protocols. The analysis is still in progress and more work will be shown in next paper. The performance of geoencryption relies on both the TESLA authentication and the receiver accuracy. Once TESLA verification is failed, the user can not proceed to the next step to compute geolock.

The performance of TESLA depends on SNR of Loran signals and authentication messages bandwidth. In Figure 16, the data is collected at Palo Alto, and Middletown is 150 km away and its SNR is approximately 30 dB. With this high signal power, the messages are successfully decoded and MACs are verified. Another set of data is collected at a different location, Los Angeles. The Middletown tower is approximately 680 km away from our receiver. In this case, not enough correct messages are obtained to perform authentication, shown in Figure 17. Without verification of MACs, the receiver fails to proceed to the next step to compute geolock; thus, the plaintext can not be decrypted.

Another important factor for the geoencryption performance is the grid space size a user specifies. This depends on the receiver accuracy. If the size chosen is too small, the user location obtained from the receiver will result in a different grid from that of the sender uses, and the random key won't be recovered because of the wrong geolock. In Figure 16, a 20 meter is used and correct geolock is computed, so the decrypted ciphertext is the same as the plaintext the sender inputs.

A different data set is taken in Palo Alto but the grid size is changed from 20 meters to 5 meters. Even though the authentication messages are verified, the receiver can't achieve an accuracy of 5 meters and the geolock computed is not correct. This results in a wrong plaintext, shown in Figure 18.

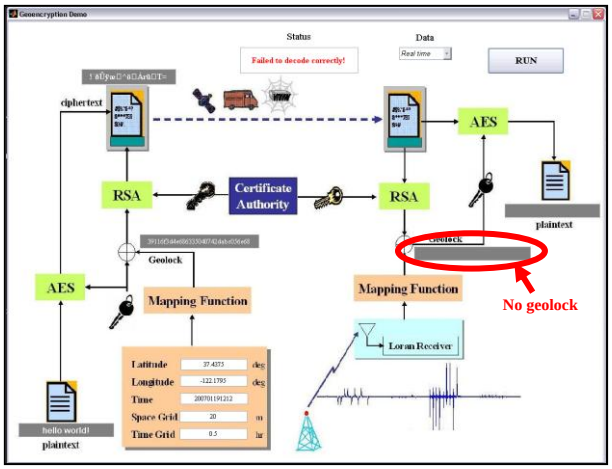

Figure 17: Receiver Location: Los Angeles

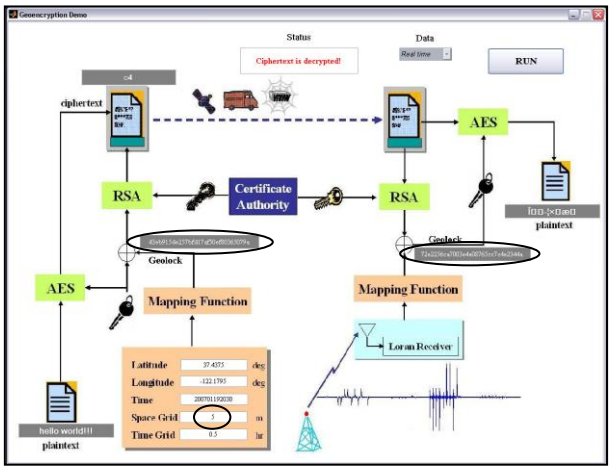

Figure 18: Receiver Location: Palo Alto

CONCLUSION

Geoencryption is an approach to location-based encryption that builds on the conventional cryptographic algorithms and protocols. It allows data to be decrypted at a specific location.

This paper describes and develops a demonstration testbed and a MATLAB GUI for geoencryption. The protocol provides protection against location bypass. Figure 16 to 18 illustrate how the protocol works. Signal authentication is proposed to provide security on Loran signal. With proper implementation of signal authentication, the protocol provides strong protection against location spoofing.

ACKNOWLEDGEMENTS

The authors would like to thank Mitch Narins of the FAA, Loran Program Office for the necessary funds to complete

this work. We would like to thank Logan Scott for his advice and suggestions. In addition, thanks goes to Jim Shima of Symmetricon for his help collecting data using the ELRR. Finally, we also would like to thank Lt. Kirk Montgomery and USCG for their support of the Middletown tests.

REFERENCE

[1] L. Scott, D. Denning, "Location Based Encryption & Its Role In Digital Cinema Distribution", *Proceedings of IONGPS/GNSS 2003*, pp288-297.

[2] L. Scott, D. Denning, "A Location Based Encryption Technique and Some of Its Applications", *Proceedings of ION NTM 2003*.

[3] Bruce Schneier, Applied Cryptography, John Wiley & Sons, Inc. 1996.

[4] International Loran Association (ILA), "Enhanced Loran (eLoran) Definitions Document", January 2007. Available at the ILA website (http://www.loran.org/)

[5] A. Perrig, R. Canetti, J.D. Tygar, D. Song, "The TESLA Broadcast Authentication Protocol", CryptoBytes, 5:2, Summer/Fall 2002, pp. 2-13

[6] B. Peterson, A. Hawes, K. Shmihluk, "Loran Data Channel Communications using 9th Pulse Modulation".

[7] K M. Carroll, A, Hawes, B. Peterson, K. Dykstra, P. Swaszek, S. Lo, "Differential Loran-C". Proceedings of European Navigation Conference GNSS 2004.

[8] S. Lo, "Broadcasting GPS Integrity Information Using Loran-C". Ph.D. Thesis.