Automated Cryptanalysis of Plaintext XORs of Waveform Encoded Speech

L. A. Khan and M. S. Baig

Abstract— Keystream reuse also known as the "two time pad" problem in case of stream ciphered data has been the focus of cryptanalysts for several decades. All heuristics presented so far assume the underlying plaintext to be uncompressed text based data encoded through conventional encoding mechanisms such as ASCII Coding. This paper presents the use of hidden Markov model (HMM) based automatic speech recognition (ASR) approach to cryptanalysis of stream-ciphered waveform-encoded speech in a keystream reuse situation. We present that an adversary can automatically recover the digitized speech signals from their plaintext XORs obtained from two different speech signals stream ciphered with the same keystream. The proposed technique can be practically employed with the existing HMM based probabilistic speech recognition techniques with some modification in the selection of HMMs, their training and the maximum likelihood decoding procedures. Simulation experiments using such modified speech recognition tools have been presented.

Index Terms— **cryptanalysis, keystream reuse, speech coding, stream cipher, two time pad.**

I. INTRODUCTION

In a stream cipher, a message *m* is exclusive ORed with a keystream *k* to produce the ciphertext *c* i.e. $m \oplus k = c$. If the keystream *k* is random and is of the same size as that of the message *m* then the stream cipher becomes a "one time pad" which is considered as a perfect cipher [1] in the cryptographic community. If two different plaintexts m_1 and m_2 are encrypted with the same keystream *k* then their results $m_1 \oplus k$ and $m_2 \oplus k$ can be XORed to neutralize the effect of the keystream *k*, thereby obtaining $m_1 \oplus m_2$. The key reuse problem in stream ciphers and its exploitation in different scenarios for uncompressed text based data have been studied since long. It has also been mentioned in the literature as the "two time pad" problem [2]. The vulnerability of keystream reuse exists with many practical systems which are still in use such as Microsoft Office [2, 3], 802.11 Wired Equivalent Privacy (WEP) [4], WinZip [5], Point to point tunneling protocol (PPTP) [6] etc. In addition to this, the two time pad problem is predicted to remain there for quite some time in the near future also because of the endorsement of counter mode of *AES* by *NIST* [2, 7] for high speed data transfer applications. In this case, cryptographers, who would have otherwise used a block cipher with *cipher block chaining (CBC)* mode, are compelled to use *AES* in the counter mode, thereby turning a block cipher into a stream cipher and the chances of reusing key streams is further enhanced. Also, there is a compelling need for a cipher mode of operation which can efficiently provide authenticated encryptions at speeds of 10 gigabits/s and is free of intellectual property restrictions. The counter mode of operation of a block cipher (e.g. AES) has been considered to be the best method for this purpose [8, 9]. This has further increased the possibility of keystream reuse in actual systems because an effective and secure key management has still been an uphill task for the crypto designers as mentioned in [10] as: *"If you think you know how to do key management, but you don't have much confidence in your ability to design good ciphers, a one-time pad might make sense. We're in precisely the opposite situation, however: we have a hard time getting the key management right….. And almost any system that uses a one-time pad is insecure. It will claim to use a one-time pad, but actually uses a two-time pad (oops)."*

As regards to hidden Markov models (HMMs), these are very rich in mathematical structure and form the theoretical basis for use in a broad scope of applications, particularly in machine recognition of speech [11]. Most contemporary speech recognizers are based on HMMs. Speech is digitized, encrypted and sent between two parties in many situations. Encryption schemes particularly designed for speech whether these are analog speech scramblers (e.g. [12]) or modern digital selective speech encryption techniques (e.g. [13]), have been the focus of security professionals since long. With the advancement in the speech digitization and compression techniques, the speech signal is now treated as an ordinary data stream of bits as far as encryption is concerned. But the acoustic and articulatory features of speech signals exploited by the automatic speech recognition (ASR) equipment especially in the distributed speech recognition (DSR) scenario [14] and automatic transcription of conversational speech [15] have encouraged us to look at their characteristics from the cryptanalytic point of view in a keystream reuse situation. We have extended the natural language approach from automated cryptanalysis of encrypted text based data to the digital data extracted from the underlying verbal conversation. An

Manuscript received 05 May, 2008.

L. A. Khan is with Information Security Department, College of Telecommunication Engineering, National University of Sciences and Technology (NUST), Rawalpindi, Pakistan.

M. S. Baig is with Center for Cyber Technology and Spectrum Management (CCT&SM), National University of Sciences and Technology (NUST), Islamabad, Pakistan. Phone: +92-51-2103420/9266480.

(e-mail: liaquatalikhan@gmail.com, msbaig-cctsm@nust.edu.pk.)

interesting by product of our attack is that it would not only decipher the information but would automatically transcribe it during the process of cryptanalysis through speech recognition.

The rest of the paper is organized as follows: In section 2, we present some background information on speech coding and hidden Markov model based speech recognition. In section 3, we discuss the previous and related work on the keystream reuse problem as well as the use of HMMs in cryptology. Section 4 presents the assumptions and concept behind our cryptanalysis technique. In section 5, we present the implementation procedure which we adopted along with experimental results. Section 6, concludes the paper and gives directions for future work.

II. Background Information

In this context, we first discuss speech coding and then automatic speech recognition with relevance to our work.

A. Speech Coding

In waveform encoding, the speech signal waveform is sampled, quantized (and compressed) and then digitally encoded. The A-law and µ-law algorithms [16] used in traditional Pulse Coded Modulation (PCM) digital telephony can be seen as very early precursors of speech digitization based on waveform encoding. In case of parameter encoding speech is considered as a source filter model in which the parameters of the model alongwith excitation information in the form of voiced/unvoiced signals is used for digital represenation of speech. In hybrid coding which has become the most popular in modern speech coding, the excitation information is not only segregated as vioced or unvoiced but the details of excitation information such as pitch, pulse positions/signs and gains are used for its representation. The common coding technique in this domain is Code Excited Linear Prediction (CELP) [17]. Although speech coders based on CELP are well known and common coders used in voice over IP networks and predominantly used in PC based systems yet some of the leading IP phone vendors unfortunately stopped supporting some implementations of CELP. This leads to G.711, which is based on waveform coding, as the common coder for PC to IP phones [18]. Moreover, most of the telecommunication links still use the A-law and µ-law algorithms of waveform coding for speech digitization. Keeping this fact in mind, we have developed our algorithm for the keystream reuse exploitation of speech signals based on waveform encoding.

B. Automatic Speech Recognition

The purpose of speech recognition is to convert spoken words to machine readable input. Two main techniques of speech recognition presently exist, one based on dynamic time warping (DTW) and the other based on hidden Markov models (HMMs). The technique which is the most common is based on hidden Markov Models and is also applicable in our scenario. A Markov process is a stochastic process in which the conditional probability of the future states depends only on the present state and not on any past state, whereas, a hidden Markov model is a statistical model in which the system to be modeled is assumed to be a Markov process with unknown parameters and the challenge is to determine the hidden parameters from the observable ones. For complete details of the hidden Markov models and their applications in speech recognition the reader is referred to [11]. All modern speech recognition tools use this technique because of its robustness, flexibility and efficiency. The goal of any ASR system is to find the most probable sequence of words $W = (w_1, w_2, w_3, \ldots\ldots)$ given an acoustic observation $O = (o_1, o_2, o_3, \ldots.o_T)$. Mathematically,

$$\hat{W} = \arg\max_{i \in L} P(w_i / O) \tag{1}$$

where L indicates the phonetic units in a language model. Equation 1 cannot be solved directly but using Baye’s Rule, the above equation can be modified as

$$\hat{W} = \arg\max_{i \in L} \frac{P(O/w_i)\,P(w_i)}{P(O)} \tag{2}$$

or it can also be written as

$$\hat{W} = \arg\max_{i \in L} P(O/w_i)\,P(w_i) \tag{3}$$

here, $P(O/w_i)$ is calculated using HMM based acoustic models, whereas $P(w_i)$ is determined from the language model.

III. Previous and Related Work

A. Keystream Reuse Exploitation

Key stream reuse vulnerability exploitation of stream ciphers dates back to the National Security Agency’s *VENONA* project [19, 20] which started in 1943 and did not even finish uptil 1980. Other worth mentioning works on the topic are those of Rubin in 1978, who for the first time formalized the process of keystream reuse exploitation [21]; Dawson and Neilson in 1996, who automated the process of cryptanalysis of plaintext XORs [22] and the recent cryptanalysis of two time pads by Joshua Mason and coauthors in 2006 [2]. Mostly the keystream reuse exploitation discussed previously is with respect to the textual data and mainly based on heuristic rules for obtaining the two plaintexts m_1 and m_2 from $m_1 \oplus m_2$ except for [2] which uses statistical finite states language models and natural language approach. Prior works also exist on automated cryptanalysis of analog speech scramblers (e.g. [23]), but no previous work exists on the use of modern automated speech recognition (ASR) techniques based on hidden Markov models (HMMs) being used for cryptanalysis of the two time pad problem for the digitally encoded speech signals. In [2], the concepts borrowed from the natural language and speech processing communities are used for text based data whereas we use similar concepts with addition of speech recognition for speech based data.

B. Use of HMMs in Cryptology

As regards to the use of hidden Markov models in cryptology, these have recently been used for several problems in this area.

The most prominent are the works of A. Narayanan and V. Shamtikov who used hidden Markov models for improving fast dictionary attacks on human memorable passwords [24]; D.X Song , D. Wagner and X. Tian who used HMMs for timing attacks on Secure Shell (SSH) [25]; D. Lee gave the concept of substitution deciphering of compressed documents using HMMs [26]; L. Zhuang, F. Zhou and J. D. Tygar modeled the keyboard acoustic emanations as HMMs [27]; C. Karlof and D. Wagner used HMMs for modeling countermeasures against side channel cryptanalysis [28] and finally the most relevant work of Joshua Mason et al [2] who used the Viterbi beam search for finding the most probable plaintext pairs from their XOR in case of textual data. It is worth mentioning here that the works involving cryptanalysis with the aid of HMMs presented so far either do not relate to two time pad cryptanalysis or pertain only to text based data. Our algorithm for cryptanalysis of the plaintext XOR of the digitized speech signals using hidden Markov model based speech recognition techniques is the first of its kind according to our knowledge and has showed encouraging preliminary results.

IV. Proposed Approach

Before discussing the concept behind our approach for exploiting keystream reuse in stream ciphered digitized speech signals, we first elaborate and justify our assumptions.

A. *Assumptions*

We assume that the cryptanalyst knows before launching an attack, the details of the speech digitization and encoding before being stream ciphered. As an a priori knowledge the cryptanalyst must know whether the audio data which he targets is waveform encoded or parameter encoded. This assumption becomes realistic from the fact that the audio encodings follow some standards and by mere knowing the standard reference, the bit level details can be easily accessed as these details are publicly available. For example, in the waveform coding we may have ITU-T G. 711 [16], the bit level details of which are publicly available. Had these details not been available then even the plaintext speech without encryption would have not been possible to be decoded. We also assume that the cryptanalyst has a priori knowledge of the language and *genre* of the underlying speech signal. For example, we may like to model military telephonic conversations, corporate discussions, and informal chat over mobile telephones, etc. In case of *VENONA* project [20], the NSA knew before hand that the said link carried Soviet military and diplomatic communication. Moreover, keeping in view the Kerckhoff's principle re-asserted by Claude Shannon as *the enemy knows the system*, the language and encoding details of the underlying speech signals can be rightly assumed to be known to the cryptanalyst before hand.

B. *The Concept*

Our method of cryptanalyzing the speech signals being encrypted with the same keystream is based on the hidden Markov model based speech recognition techniques. All modern speech recognition tools use this technique because of its robustness, flexibility and efficiency [11]. The three basic questions with respect to the HMMs and their solutions as regards to speech recognition are effectively utilized with some modification in our case. The three basic problems of interest that are to be solved for the model to be useful as regards to the cryptanalysis of speech signals encrypted with the same key are:

1) *Finding Probability of Observation given a Model*

Given an observation sequence *O* (XORed ciphered speech waveforms in our case) and a model $\lambda = \left(\pi_i, A, B\right)$ where π_i is the initial probability of states (XORed phonemes in our case), *A* is the transition probability of states and *B* is the emission probability distribution of the observation sequence, how do we compute the probability that the given sequence of observations was produced by the model λ i.e. $P\left(O/\lambda\right)$? The solution to this problem allows us to choose the model which best matches the observation sequence which in our case would be a sequence of XORed speech samples.

2) *Finding Internal States given Model and Observation*

Given the observation sequence *O* of XORed speech waveforms and the model λ , how do we choose a corresponding sequence of states i.e. XORed spoken words in case of isolated word recognizer and sequence of XORed phonemes in case of continuous speech recognizer, which is optimal and best explains the observation? This is the problem in which we try to find the hidden part of the model i.e. to find the "correct" state sequence. In this case we have to impose certain optimality criterion like the sequence with maximum log probability may be selected.

3) *Adjusting Model Parameters given Observation*

How do we adjust the model parameters $\lambda = \left(\pi_i, A, B\right)$ to maximize the probability of the observation sequences of XORed speech waveforms given the model λ ? This is the training part of the models and on one hand is the most crucial part in the sequence of events but on the other hand gives a lot of flexibility to the cryptanalyst thus empowering him to adjust his model for all sorts of varying situations like different languages, accents, different speech coding and compression techniques and even different noisy conditions. This allows the crypanalyst to create best models for real phenomena.

All the abovementioned problems and their efficient mathematical solutions have a very rich literature with respect to speech recognition [11]. In the conventional speech recognition techniques, the hidden Markov models are trained for complete words in case of isolated word recognizers and for phonemes in case of continuous speech recognizers. In our case, for isolated word recognition, we have to first list down all the possible combination of words resulting from the XOR of the two speech signals and hence the HMMs required to be trained will increase from *n* to n^2. Since the list of words is generally very large, therefore, this approach of training the HMMs would be very computational intensive and maybe impractical. A better and more efficient approach is to train the

HMMs for the exclusive ORed pairs of all the possible phonemes in the language under test. In this case, since the number of phonemes is relatively very small as compared to the number of possible words, the increase in computational complexity from n to n^2 does not become unachievable. For example, in English language there are about *40* to *50* phonemes and hence the number of HMMs to be trained in this case would be at the most *2500* (50^2) which are not high as regards to the computational resources available to a normal user these days. Using this approach would not require any major modification in the conventional phoneme based speech recognition procedure. In the pre-computation phase which comprises of selection and training of HMMs, we will first list down all the possible phonemes in the language and then we will pair each phoneme with every other phoneme including itself in the list. We will then assign each HMM to each phoneme pair and then train it with the training data selected on the basis of the a priori knowledge about the language and encoding mechanism of the speech signal. Once the HMMs corresponding to phoneme pairs are fully trained then these can be used for the identification of phoneme pairs in a given sequence of XORed speech samples in a process similar to the decoding part of a conventional HMM based ASR system. The decoded phoneme pairs are first separated into two distinct groups and then the phonemes within a group can then be and combined to form different words and sentences. For both these steps, help can be taken from the semantics and syntactic rules of the language.

V. Implementation of Proposed Approach

The implementation part of our attack involves a pre computation phase which involves selection and training of the models and then the decoding phase which corresponds to the recognition of sequence of phoneme pairs which gives the highest log probability. Both the phases are interrelated and interdependent and the accuracy of the attack is greatly dependent on how well these two parts of the attack are carefully employed and joined.

A. HMM Selection and Training Phase

This phase corresponds to the pre computation part of the speech recognition in which the HMMs are first selected and then trained with the help of speech samples available with respect to each phoneme pair. This is done once for a particular language and specific speech encoding procedure. In order to prove the concept, we present a simple example in which we take two phonetically balanced English sentences: *Clothes and lodging are free to new men*; and *All that glitters is not gold at all*. We bit wise XORed the digital encoded forms of these sentences to simulate the keystream reuse scenario. Fig. 1(a), (b) show the spectrogram and waveform of the two sentences along with their transcription. The transcription at the phoneme level is obtained from the British English pronunciation dictionary BEEP [29]. For simplicity of implementation the silence between words is not marked separately, only the initial silence and the end silence are marked. Fig 1(c) corresponds to the bitwise XOR of the two signals and the associated transcriptions in the form of phoneme pairs. The "+" sign in the figure corresponds to XOR. For the individual sentences the number of phonemes is 27 for sentence 1 and 26 including silence (*sil*) and hence the HMMs required to be trained for the XOR case would be 702 (27x26) at the max. These are obtained by pairing every phoneme of sentence 1 with every phoneme of sentence 2. We used ten utterances each of the sentence 1 and sentence 2 from ten different speakers, labeled the *wave* files and then bit wise XORed both the files again labeling these with the HMM boundaries clearly defined. For recordings, transcription and speech recognition we used the *HTK* [30] which is a toolkit based on C language for analysis of hidden Markov models particularly for machine recognition of speech. The recordings and transcription can be obtained by the *HTK* tool *HSLab*. The above mentioned acoustical events were modeled by 167 HMMs with each HMM corresponding to one XORed pair of phonemes. Since all the possible phonemes do not occur hence the actual number (167) of phoneme pairs is quite less than the total possible number (702). The basic design of the HMM we used in this case for all the models is as shown in Fig. 2. Since speech recognition equipment cannot process waveforms directly, these are to be converted into more compact form. The configuration we used for the speech recognition was based on Mel Frequency Cepstral Coefficients (MFCC) [31] with 12 first MFCC coefficients, the null MFCC coefficient which is proportional to the total energy in the frame, 13 Delta coefficients estimating the first order derivative of MFCC coefficients and 13 acceleration coefficients estimating the second order derivatives, altogether a 39 coefficient vector is extracted from each signal frame. The frame length is 25 milliseconds with 10 milliseconds frame periodicity. The parameters which are to be estimated for each HMM during the training phase are transitional probabilities a_{ij} and the single Gaussian observation function for each emitting state which is described by a mean vector and variance vector (the diagonal elements of the autocorrelation matrix). In our case we have to estimate all these values for each of the 167 HMMs during the training phase. The *HTK* tools *HInit*, *HCompV,* and *HRest* can be used for this purpose.

Before using our HMMs we have to define the basic architecture of our recognizer. In actual case this depends on the language and the syntactic rules of the underlying task for which the recognizer is used. We assume that these things like the language of the speakers and the digital encoding procedures are known to the cryptanalyst before hand. *HTK*, like most speech recognizers, works on the concept of recognition network which are to be prepared in advance from task grammars, and the performance of the recognizer is greatly dependent on how well the recognition network maps the actual task of recognition. In addition to the recognition network, we need to have a task dictionary which explains how the recognizer has to respond once a particular HMM is identified. The task grammar for our recognition network is shown in Fig. 3. The recognition network for our experiment is shown in Fig. 4. The *HParse* tool of *HTK* can be used to construct the recognition network from the task grammar. *HSGen* can be used for testing and verification of the recognition network.

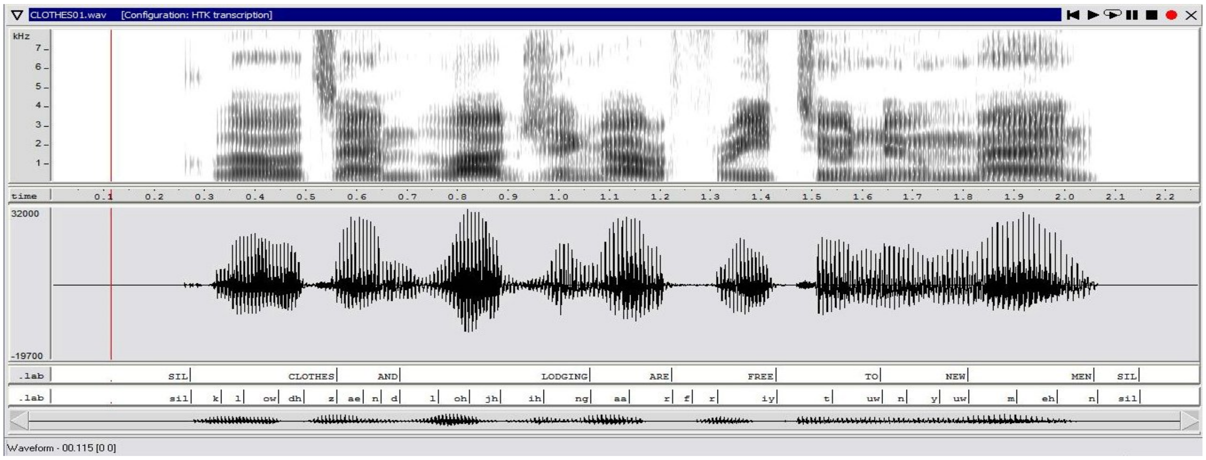

CLOTHES01.wav [Configuration: HTK transcription]
kHz
7
6
5
4
3
2
1
time
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2
32000
-19700
.lab
SIL CLOTHES AND LODGING ARE FREE TO NEW MEN SIL
.lab
sil k l ow dh z ae n d l oh jh ih ng aa r f r iy t uw n y uw m eh n sil
Waveform - 00.115 [0 0]

Fig. 1(a). Spectrogram and waveform along with transcription of the sentence:
Clothes and lodging are free to new men.

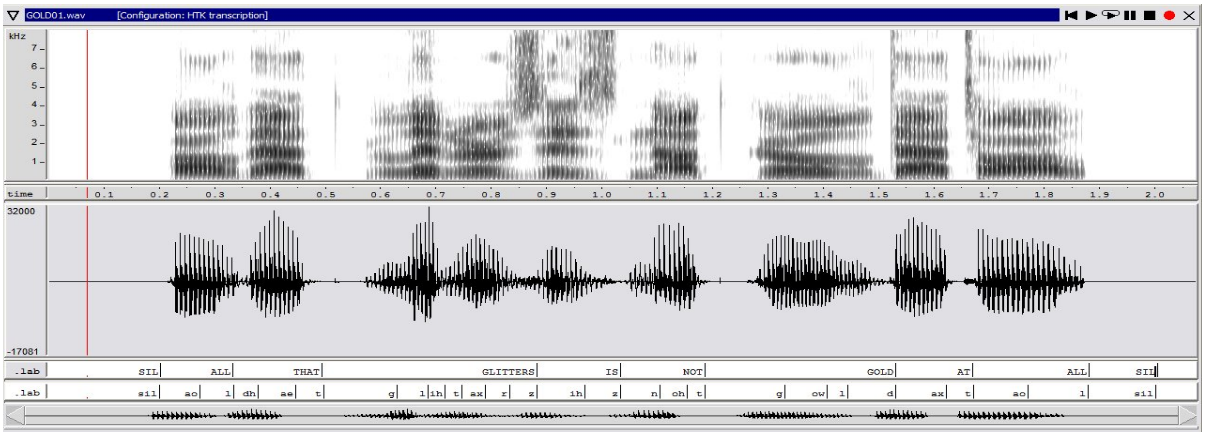

GOLD01.wav [Configuration: HTK transcription]
kHz
7
6
5
4
3
2
1
time
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0
32000
-17081
.lab
SIL ALL THAT GLITTERS IS NOT GOLD AT ALL SIL
.lab
sil ao l dh ae t g l ih t ax r z ih z n oh t g ow l d ax t ao l sil

Fig. 1(b). Spectrogram and waveform along with transcription of the sentence:
All that glitters is not gold at all.

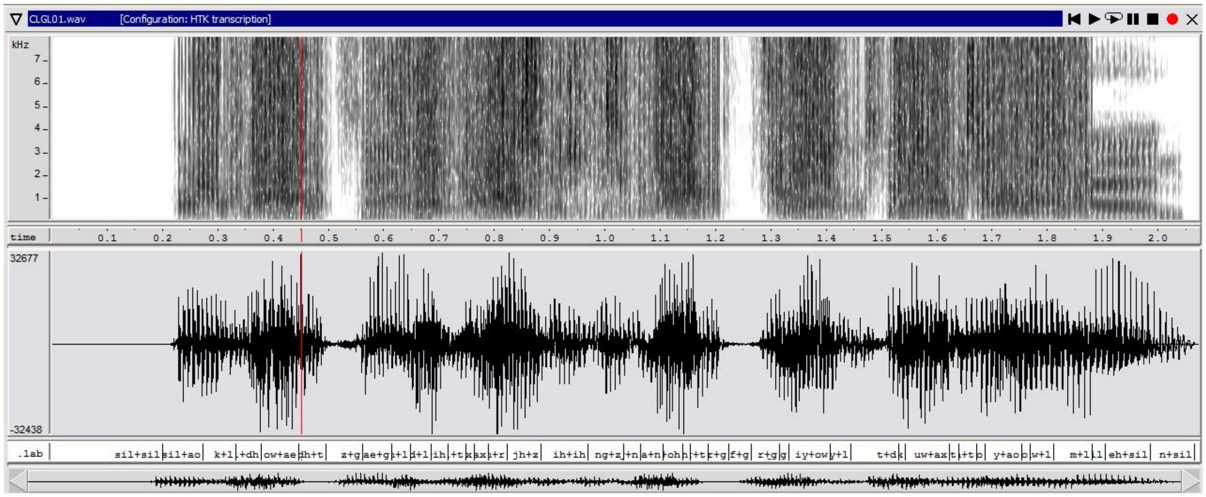

CLGL01.wav [Configuration: HTK transcription]
kHz
7
6
5
4
3
2
1
time
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0
32677
-32438
.lab
sil+sil sil+ao k+l l+dh ow+ae dh+t z+g ae+g n+l d+l ih l+t x ax h+r jh+z ih+ih ng+z +n a+n r+oh h+t r+g f+g r+g g iy+ow y+l t+d k uw+ax t +t o y+ao o w+l m+l il eh+sil n+sil

Fig. 1(c). Spectrogram and waveform along with transcription of XOR of the sentences.

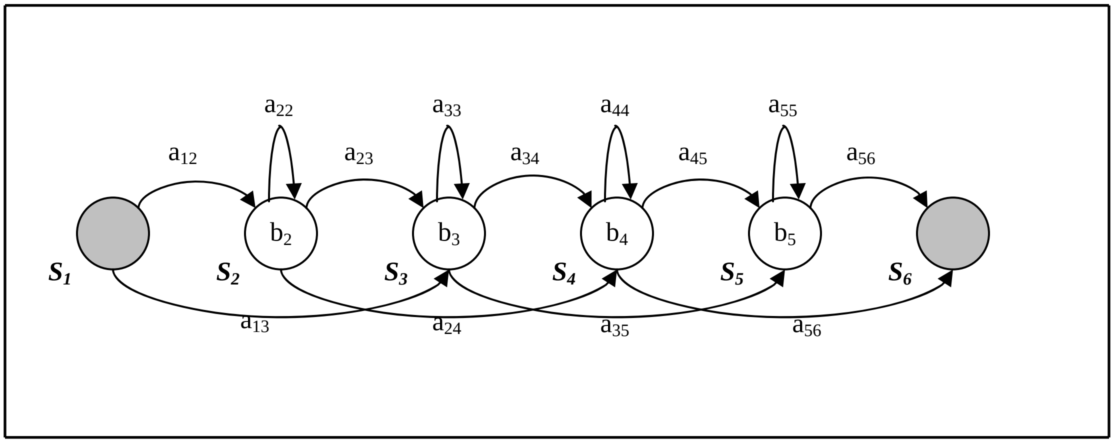


Fig. 2. Basic Topology of the HMM

```
/* Task grammar*/
$WORD = sil+ao | k+l | l+dh | ow+ae | dh+t | z+g | ae+g | n+l  | d+l | l+ih | l+t | l+ax | oh+ax | oh+r | jh+z |ih+ih
| ng+z | ng+n | aa+n | r+oh | r+t | r+g | f+g | iy+g | iy+ow | iy+l | t+d | t+ax | uw+ax | uw+t | n+t | n+ao | y+ao |
uw+ao | m+l | m+sil | eh+sil | n+sil | k+ao | ow+dh | ow+t | dh+g | z+l | ae+l | ae+ih | n+ih | d+t | l+r | oh+z | jh+ih
| ih+z | aa+t | f+ow | r+ow | r+l | iy+d | iy+ax | iy+t | t+t | y+l | uw+sil | sil+l | f+oh | f+t | n+ax | y+t | m+ao | eh+l
| l+ao | ow+ao | dh+ao | ae+dh | ae+ae | d+g | l+g | oh+l | jh+l | ng+t | aa+ax | r+ax | r+r | f+r | f+z | f+ih | r+ih | iy+z
| t+z | uw+n | n+oh | uw+g | uw+ow | m+ow | eh+ow | n+d |  sil+ax | sil+t | n+ae | d+ae | l+ae | oh+ae | oh+t | oh+g
| jh+g | ih+l | ng+l | aa+ih | f+ax | r+z | iy+ih | uw+z | n+n | y+oh | m+g | eh+g | n+ow | sil+ow | sil+d | l+l | ow+l
| dh+l | z+dh | jh+t | ih+t | ng+g | aa+g | f+l | n+z | y+z | m+z | eh+n | eh+oh | sil+oh | dh+dh | ih+g | aa+l | t+ih |
uw+ih | m+n | sil+g | k+sil | n+dh | ng+ax | aa+r | aa+z | f+n | l+sil | ow+sil | z+ao | ih+ax | ng+r | r+n | iy+oh | r+d
| ae+t | oh+ih | jh+ax | ih+r | t+g | y+d | uw+d | m+ax | eh+ax | aa+oh ;

( [ START_SIL ] { $WORD } [ END_SIL ] )
```

Fig. 3. Task Grammar for the Recognition Network

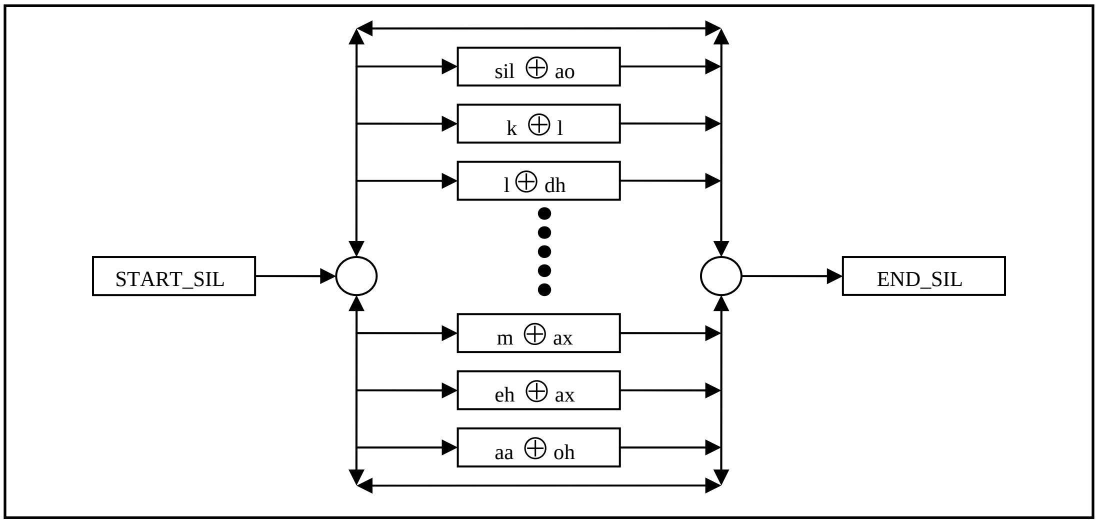
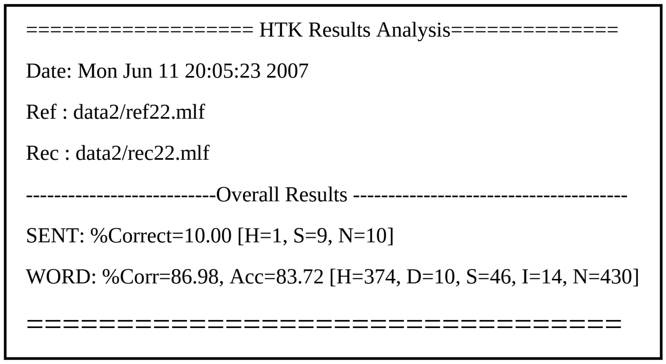


Fig. 4. Recognition Network

A. Decoding Phase

Once the pre-computation phase has carefully been completed, the decoding process becomes pretty simple and elegant. An input speech signal, which in our case will be the bitwise XOR of two unknown speech waveforms, is first converted to a sequence of *n* MFCC vectors and is then fed as input to the recognizer. Every path from the start node to the end node in the recognition which passes through precisely *n* emitting states is a prospective recognition hypothesis. Each of these paths has a log probability which is computed by summing the log probability of individual transition in the path and the log probability of each emitting state generating the corresponding XORed vector. Within the model, transitions are determined from the model parameters (a_{ij}), between two models the transitions are regarded as constant and in case of large recognition networks the transition between end words are determined by language models likelihoods attached to the word level networks. The decoder lists those paths through the network which have the highest log probability. These paths are found using a *Token Passing Algorithm* [30]. At time 0, a token is placed in every possible start node. Each time step, tokens are propagated along connected paths through the recognition network stopping whenever they hit an emitting HMM state. When there are more than one out going paths from a node, the token is copied so that all possible paths are explored in parallel. As the token passes across transitions the corresponding transition and emission probabilities add up to its log probability. The token also maintains a history of its route during the process of propagation. In a large network, which will definitely be in our the case, a *beam search* may be used in case of which a record of the *best token* overall is kept while deactivating all tokens whose log probability falls more than a *beam width* below the best. This *beam search* technique has one problem i.e. if the pruning beam width is set too small then the actual recognition path might be pruned before its token reaches the end of the observation i.e. it may result in a *search error*. Setting the beam width is thus a compromise between computational load and avoiding *search errors*. Fortunately, *HTK* tools *HVite* takes care of all these speed and computation problems [30]. The initial experiments which we performed gave us up to 80 percent correct recognition of phoneme pairs obtained though *HResults* tool of *HTK* as shown in Fig. 5.



```
================== HTK Results Analysis==============

Date: Mon Jun 11 20:05:23 2007

Ref : data2/ref22.mlf

Rec : data2/rec22.mlf

---------------------------Overall Results ---------------------------------------

SENT: %Correct=10.00 [H=1, S=9, N=10]

WORD: %Corr=86.98, Acc=83.72 [H=374, D=10, S=46, I=14, N=430]

=================================
```

Fig 5. HTK Recognition Performance

B. Detailed Experimentation

After getting encouraging results from the experiment in the controlled environment, we then tested our technique on actual speech files. For this purpose, we selected the *Switchboard* Corpus [32] which is a collection of telephone bandwidth conversational speech data collected from T1 Lines. The speech files are fully transcribed. The reason for this selection was to simulate the situation of eavesdropped encrypted communication of waveform encoded speech. In order to simulate the keystream reuse scenario, we selected 256 speech files and XORed them with each other. The acoustical events were modeled with 588 HMMs with each HMM corresponding to one pair of phonemes. Since all the possible phonemes do not occur always hence the actual number of phoneme pairs (588) is quite less than the total possible number (2500). As discussed earlier, speech recognition tools cannot process speech waveforms directly. Different acoustic feature representations were used for recognition purposes. For all the representations, we used frame length of 25 milliseconds with 10 milliseconds frame periodicity. The parameters which were to be estimated for each HMM during the training phase were transitional probabilities a_{ij} and the single Gaussian observation function for each emitting state which is described by a mean vector and variance vector (the diagonal elements of the autocorrelation matrix). The different acoustic features extracted for recognition purposes include linear predictive coefficients, linear predictive reflection coefficients, linear predictive Cepstral coefficients and Mel frequency Cepstral coefficients along with delta and reflection coefficients. For testing purpose, we selected twenty different files from the Switchboard corpus not included in the training data, XORed these to get ten files. As an initial test we also selected ten different files from the training data and fed these to the recognizer. The experimental results with respect to test files selected from the training data as well as arbitrary test files are depicted in Table I. The best accuracy results were presented by the Mel Frequency Cepstral Coefficients (MFCC) with delta and acceleration coefficients for both the test file categories.

TABLE I : RECOGNITION ACCURACIES OF DIFFERENT ACOUSTIC FEATURES

SNo	Feature Extraction Mechanism	Recognition Accuracy (%)	
		Training Data Test Files	Arbitrary Test Files
1.	Linear Predictive Coefficients	65.93	29.51
2.	Linear Predictive Reflection Coefficients	69.06	28.61
3.	Linear Predictive Cepstral Coefficients	72.09	34.62
4.	Mel Frequency Cepstral Coefficients (MFCC)	74.72	40.73
5.	Linear Predictive Cepstral + Delta Coefficients	77.15	37.81
6.	Mel Frequency Cepstral + Delta + Acceleration Coef.	79.96	59.09

C. *Complexity Analysis*

For the complexity of the training phase, the increase in the number of models to be trained is from n to n^2 at the most, if we pair each phoneme with every other phoneme and of course itself. For the decoding phase, the number of phonemes which are to be checked at each stage of the decoding path is also increased from n to n^2 and the number of possible paths increases from n^2 at each stage to n^4 at each stage. Hence we may expect an exponential increase in the decoding from the conventional speech recognition. One way to reduce the number of iterations at each stage is to carry out beam search in which the width of the beam should be carefully selected in order to avoid pruning of useful paths at the early stage of the recognition network. Fortunately, *HTK* supports the beam viterbi search approach and hence can be employed in our case with no major modification in the decoding phase.

VI. Conclusions and Future Work

In this paper, we presented that how the keystream reuse problem of stream ciphers can be exploited in case of waveform encoded speech signals. Prior to this work, one could safely reuse keystreams in case the underlying plaintext data was speech as all exploitation techniques presented before this transaction assumed the underlying plaintext to be uncompressed text-based data encoded through conventional encoding techniques such as ASCII coding. We have shown that the conventional speech recognition techniques can be adapted to be used for cryptanalysis of two time pads in case of stream ciphered digitized speech signals. *HTK* and other automatic speech recognition (ASR) tools can be effectively modified for this purpose. These speech recognition tools work on the concept of context-dependent tied-state multi-mixture tri-phones [30] which make the performance of the recognizer more flexible and robust. The use of tri-phones in the keystream reuse scenario needs to be looked into in the future assignments. The parameter encoded and compressed speech signals in the keystream reuse situation can also be looked into as a future work.

References

[1] C.E. Shannon. *A mathematical theory of communication*. Bell System Technical Journal, 27:379-423, July, 1948.

[2] Joshua Mason, Kathryn Watkins, Jason Eisner and Adam Stubblefield. *A natural language approach to automated cryptanalysis of two time pads.* In 13th ACM Conference on Computer and Communications Security, Nov, 2006.

[3] H. Wu. *The misuse of RC4 in Microsoft Word and Excel,* Cryptology ePrint Archive, Report 2005/007, 2005. http://eprint.iacr.org.

[4] N. Borisov, I. Goldberg and D. Wagner. *Intercepting mobile communications: The insecurity of 802.11.* In MOBICOM 2001, 2001.

[5] T. Kohno. *Attacking and repairing the WinZip encryption scheme,* In 11th ACM Conference on computer and communications security, pp 72-81, Oct 2004.

[6] B. Schneier, Mudge and D. Wagner. *Cryptanalysis of Microsoft PPTP Authentication Extensions (ms-chapv2).* CQRE'99, 1999.

[7] M. Dworkin. R*ecommendation for block cipher modes of operations*, NIST Special Publication 800-38A, 2001.

[8] David A. McGrew and John Viega, *The Galois/Counter mode of Operation (GCM),* May, 2005. Available from http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/gcm/gcm-revised-spec.pdf.

[9] R. Housley and A. Corry, *GigaBeam high speed radio link encryption,* RFC 4705, Oct, 2006. Available from http://tools.ietf.org/html/rfc4705

[10] Bruce Schneier. *Cryptogram*-Newsletter, Oct, 2002.

[11] L. R. Rabiner. *A tutorial on hidden markov models and selected applications in speech recognition,* Proceedings of the IEEE, 77(2): 257-286, Feb, 1989.

[12] E. Dawson. *Design of a discrete cosine transform based speech scrambler,* Electronics Letters, vol. 27, pp. 613-614, Mar, 1991.

[13] Chung Ping Wu and C. C. Jay Kuo. *Fast encryption methods for audiovisual data confidentiality.* Proceedings of SPIE vol. 4209, pp. 284-295, Nov, 2000.

[14] David Pearce. *Enabling new speech driven services for mobile devices: An overview of the ETSI standards activities for distributed speech recognition front ends.* AVIOS 2000: The Speech Applications Conference, CA, USA, May, 2000.

[15] M. J. F. Gales, B. Jia, X. Liu, K. C. Sim, P. C. Woodland and K. Yu. *Development of the CUHTK 2004 RT04F Mandarin Conversational Telephone Speech Transcription System. Proc. ICASSP 2005, Volume I, pp. 841-844, March 2005.*

[16] ITU-T Recommendations G.711. *Pulse Code Modulation (PCM) of Voice Frequencies,* Nov, 1988.

[17] M. R. Schroeder and B. S. Atal, Code-*excited linear prediction (CELP): high-quality speech at very low bit rates*, in Proceedings of ICASSP, vol. 10, pp. 937-940, 1985.

[18] Randy Goldberg & Lance Riek. *A Practical Handbook of Speech Coders,* CRC Press NYC, pp 67, 2000.

[19] P. Wright. *Spy Catcher*. Viking, New York, NY, 1987.

[20] R. L. Benson and M. Warner. *VENONA: Soviet Espionage and the American Response 1939-1957.* Central Intelligence Agency, Washington D.C., 1996.

[21] R. Rubin. *Computer methods for decrypting random stream ciphers.* Cryptologia, 2(3):215-231, July 1978.

[22] E. Dawson and L. Nielsen. *Automated cryptanalysis of XOR plaintext strings.* Cryptologia, 20(2): 165-181, April, 1996.

[23] B. Goldburg, E. Dawson, S. Sridharan. *The automated cryptanalysis of analog speech scramblers*, Adnances in Cryptology, EUROCRYPT'91, Springer-Verlag LNCS 457, pp 422, April, 1991.

[24] A. Narayan and V. Shmatikov. *Fast dictionary attacks on human-memorable passwords using time-space trade-off.* 12th ACM Conference on Computer and Communications Security, pp 364-372, Washington D.C., Nov, 2005.

[25] D. X. Song, D. Wagner and X. Tian. *Timing analysis of keystrokes and timing attack on SSH*. 10th USENIX Sec. Symposium, Aug, 2001.

[26] D. Lee. *Substitution deciphering based on HMMs with application to compressed document processing.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(12): 1661-1666, Dec, 2002.

[27] L. Zhuang, F. Zhou and J. D. Tygar. *Keyboard acoustic emanations revisited.* 12th ACM Conference on Computer and Communications Security, pp 373-382, Washington, D.C., Nov, 2005.

[28] C. Karlof and D. Wagner. *Hidden Markov models cryptanalysis.* Cryptographic Hardware and Embedded Systems- CHES'03, Springer-Verlag LNCS 2779, 17-34, 2003.

[29] BEEP-*British English Pronounciation Dictionary (Phonetic Transcriptions of over 250,000 English words).* http://svr-www.eng.cam.ac.uk/comp.speech/Section1/Lexical/beep.html.

[30] S. J. Young, G. Evermann, T. Hain, D. Kershaw, G. L. Moore, J. J. Odell, D. Ollason, D. Povey, V. Valtchev and P. C. Woodland, *The HTK Book.* Cambridge University, Cambridge, 2003. http://htk.eng.cam.ac.uk/download.shtml.

[31] V. Tyagi and C. Wellekens, *On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition.* ICASSP '05. vol. 1, 2005, pp. 529–532.

[32] J. J. Godfrey, E. C. Holliman, and J. McDaniel. *SWITCHBOARD: Telephone speech corpus for research and development,* Proceedings of ICASSP, San Francisco, 1992.