The Impact of Confidentiality on Quality of Service in Heterogeneous Voice over IP Networks

Johnathan M. Reason and David G. Messerschmitt

University of California
Berkeley, Electrical Engineering and Computer Sciences Department
Berkeley, CA 94720
{reason, messer}@eecs.berkeley.edu

Abstract. With the advent of ubiquitous access to multimedia content via wireless networks, users are more likely to have their data traverse a heterogeneous internetwork. Given the open nature of the Internet and wireless links, more users will demand end-to-end confidentiality. However, because of inherent error-expansion properties of secret-key encryption, it is a challenge to provide good subjective quality and end-to-end confidentiality for multimedia data, particularly in network environments subject to both loss and corruption impairments. This paper analyzes the affect end-to-end confidentiality has on the quality of service (QoS) in Voice over IP (VoIP) networks. To satisfy a given QoS objective, we show that mitigating the error-expansion caused by confidentiality comes at a cost. We measure this cost by increased delay, reduced bandwidth, and reduced traffic capacity. Thus, for this class of applications, we motivate the need for error-robust encryption and introduce one such technique in this paper.

1 Introduction

The Internet is emerging as the network of choice to advance the convergence of computer and telephony communications, and voice over IP (VoIP) is the predominate service leading the way. However, because the Internet was not designed with real-time traffic as a target, there are still a number of challenges to providing good voice quality. In particular, the lack of quality of service (QoS) guarantees for delay, jitter, packet loss, and bandwidth affect the voice quality attainable over the Internet. The Real-Time Transport Protocol (RTP) provides some functionality suited for carrying real-time content (e.g., a sequence number and timestamp) [6], but more work is needed to provide voice quality comparable to that of the Public Telephone Switching Network (PTSN).

In recent years, many authors have contributed to the literature regarding QoS for VoIP networks [8]. Most of these articles focus on mitigation techniques for the aforementioned impairments. With recent advances in robust, header

E.S. Al-Shaer and G. Pacifici (Eds.): MMNS 2001, LNCS 2216, pp. 175–192, 2001.

compression to promote wireless IP [7], VoIP over wireless is also viable. Thus, future VoIP networks will likely be heterogeneous networks.

Heterogeneity introduces another impairment, corruption, which can further degrade voice quality. Natural phenomenon present in wireless channels (e.g., multipath fading), but not in packet-switched networks, manifest this impairment. A heterogeneous VoIP network must contend with the combination of all these impairments.

This observation raises a special QoS challenge in providing both good subjective quality and end-to-end confidentiality for VoIP. In particular, secret-key encryption techniques have inherent error-expansion properties that can degrade subjective quality. Historically, the purpose of confidentiality has been viewed solely form the perspective of its strength. Improper deployment of confidentiality can lead to weakened security, and encryption that is not secure is useless. In this paper, we show that improper deployment of confidentiality can also lead to a trade-off in the QoS targets for delay, bandwidth and traffic capacity. While the primary concern will always be the strength of confidentiality, its impact on QoS is also of importance. We motivate the need for error-robust encryption to provide confidentiality in VoIP networks and introduced one suitable technique in this paper.

2 Error-Expansion Properties of Encryption

Secret-key algorithms perform bulk encryption of sensitive data in real-time applications such as VoIP. In contrast, because of their slower performance, public-key algorithms are usually reserved for non-real-time applications, such as providing confidentiality of sensitive credentials exchanged during authentication protocols.

We classify secret-key algorithms in two categories: block and stream. Block algorithms encrypt/decrypt blocks of bits at a time and stream algorithms encrypt/decrypt a single bit at a time. In the cryptography literature, the term *cipher* describes a family or grouping of algorithms. For example, the term *block cipher* refers to the grouping of a block encryption algorithm with a block decryption algorithm for enciphering and deciphering data. On the other hand, its plural, *block ciphers*, refers to the entire family of possible groupings. We will adhere to this convention in this paper. Furthermore, we call the device that performs encryption the *encryptor* and the device that performs decryption the *decryptor*. The original messages are called *plaintext* and the encrypted messages are called *ciphertext*.

2.1 Expansion of Bit Errors

Cryptographers design block algorithms to satisfy the *avalanche effect* [1]. The avalanche effect states that an average of one-half of the output bits should change whenever a single input bit changes. Although it has not been proven that this property is a necessary condition for security, this property is exhibited by all block algorithms that have found their way into practice [2,3]. It is a desirable property because it says that each output bit must depend on all the input bits. In other words, an algorithm that has this property does not exhibit any statistical correlation between input and output that an adversary might use in an attack. The consequence of this property is that block ciphers multiply bit errors. That is, a single bit error in the ciphertext received at the input to the decryptor will result in multiple bit errors in the recovered plaintext at the output of the decryptor.

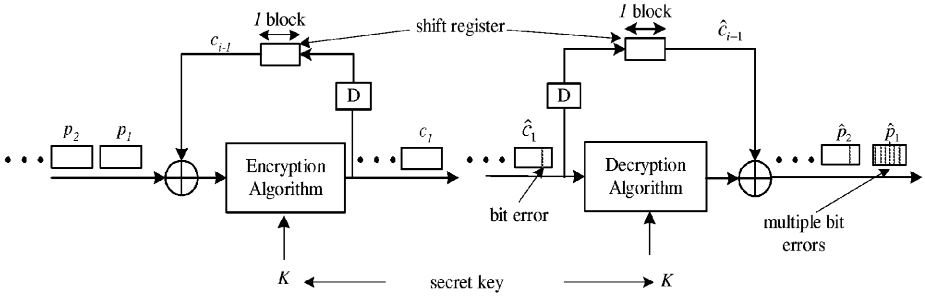

Fig. 1. Example of bit-error expansion for a block cipher in cipher block chaining (CBC) mode. Because of the avalanche effect, one or more bit errors in ciphertext block, $\hat{c}_1$, leads to multiple bit errors in the corresponding recovered plaintext block,$\hat{p}_1$.

Fig. 1 illustrates this property for a generic block cipher in cipher-block-chaining (CBC) mode. The encryptor produces N-bit ciphertext blocks then transmits them to the decryptor. During transmission, impairment along the path between encryptor and decryptor corrupts one bit of ciphertext block c_1. Because of the avalanche effect, the decryptor multiplies this error by corrupting half of the recovered bits in plaintext block p_1. Additionally, the next recovered block (p_2) will have a single corrupted bit in the same bit position as the bit error in c_1. For a VoIP service using block encryption for confidentiality, in the worst case, if there is at least one bit error in every block of ciphertext, then the recovered voice would make a persistent static sound (i.e., pseudo random noise). It it does not matter how badly a ciphertext block is corrupted; because of the avalanche effect, the decryptor will always garble the recovered plaintext block.

Because of this property, block ciphers are seldom used for real-time services in environments that have appreciable corruption (e.g., the wireless, land-mobile channel of cellular systems). However, block ciphers do perform well in environments that have appreciable loss, but negligible corruption (e.g., landline, packet-switched networks). In the case of VoIP, the latter statement is true as long as the VoIP service maintains the block framing within a voice packet.

Although there are many modes of operation for block ciphers, there are four modes that have gained wide acceptance [4]. Each of these modes has a particular error structure [5]. We highlight CBC mode because it is most often used as the default operating mode for packet communications. In particular, the Data Encryption Standard (DES) in CBC mode is specified as the default encryption method for RTP communications [6].

2.2 Propagation of Synchronization Errors

In contrast, stream ciphers do not multiply bit errors, but do propagate synchronization errors caused by insertion or deletion of bits. That is, if the decryptor loses synchronization with the encryptor, the decryptor will garble all the recovered plaintext bits after the synchronization error until the system restores synchronization.

Fig. 2 illustrates this property for a *synchronous* stream cipher: a stream cipher that requires some external mechanism to maintain synchronization between encryptor and decryptor. The encryptor produces a stream of ciphertext bits (formatted as packets) then transmits them to the decryptor. Ciphertext packet c_1 is lost in transmission, thereby causing a synchronization error. Until the encryption system restores synchronization between the encryptor and decryptor, the decryptor will produce a pseudo-random stream of bits at its output. For a VoIP service using stream encryption for confidentiality, in the worst case, if the system does not restore synchronization, then the recovered voice from the voice codec would sound like static.

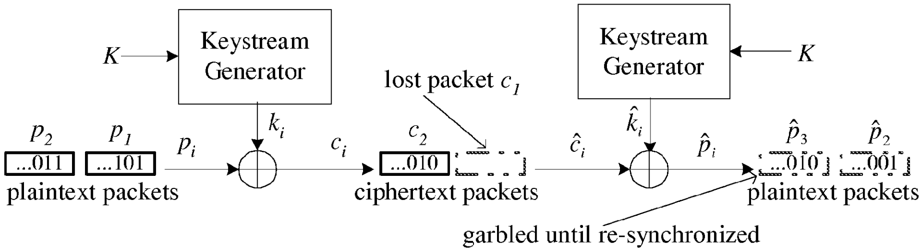

Fig. 2. Propagation of a synchronization error for a synchronous stream cipher. A lost ciphertext packet will cause loss of synchronization between encryptor and decryptor. The decryptor will garble all recovered plaintext packets ($\hat{p}_2, \hat{p}_3, ...$) until re-synchronization.

Because of this property, environments where packet re-ordering is common and packet loss is appreciable confound the task of re-synchronization. Packet-switched networks exhibit these characteristics, particularly when data is transported using a real-time, unreliable, connection-less transport protocol (e.g., RTP/UDP). As a result, VoIP applications that provide confidentiality opt to avoid the synchronization issues of stream ciphers by exclusively using block ciphers. Stream ciphers do perform reasonably well over connection-oriented transport, where there might be appreciable loss and/or corruption, but no re-ordering of packets. Wireless links exhibit this combination of impairments. However, stream ciphers used in these environments often derive their synchronization means from physical-layer parameters that are not available to higher layers. For example, Bluetooth derives its cipher's synchronization from the master clock [9]. Moreover, the absence of packet re-ordering over the wireless link allows the use of heuristics to make reasonable guesses at the cipher synchronization bits, in the event they are corrupted by the channel.

2.3 Heterogeneity

In a heterogeneous network that includes wireless links, it is not clear which type of cipher to use for end-to-end confidentiality. In such an environment, the end-to-end path is subject to loss, corruption, and re-ordering of packets. Because of corruption, block ciphers are not well suited for this scenario. Similarly, the combination of these impairments exacerbate the synchronization problems of stream ciphers. In particular, many systems implement synchronization by providing cipher synchronization bits in each packet. As we show in Section 4, this method works fine in the absence of corruption, but when corruption is present this technique leads to degraded QoS. Thus, for VoIP over wireless, end-to-end confidentiality is a challenge.

3 Why End-to-End?

Confidentiality can be deployed on a link-by-link basis or on an end-to-end basis. For link-level confidentiality, each end-system and intermediate node (e.g., bridges, routers, and gateways) of each link performs encryption and decryption of each packet passing through that node. Whereas for end-to-end confidentiality, only the end-systems perform the encryption and decryption operations.

Link encryption is effective for point-to-point communications, where the two end-systems communicate directly with each other over a single link. However, this level of confidentiality does not scale well over many links and many networks. For example, over an end-to-end path that consists of N secure links, each packet would be encrypted and decrypted N times. Thus, link encryption adds processing delay that grows linearly with the number of secure links over the end-to-end path. This increased delay can adversely affect the QoS in VoIP networks. In contrast, for end-to-end confidentiality, each packet is only encrypted and decrypted once, regardless of how many links comprise the end-to-end connection. Link encryption might also be effective when there are only a small number of untrusted links in the end-to-end path. Cellular telephone communications follow this scenario, where the wireless link between the base station and the cellular telephone is the only encrypted link. Voice data is sent in the clear over the backbone network (i.e., PTSN), which is assumed to be a trusted network. However, one can debate the validity of this assumption. Anyone with the right equipment can eavesdrop on a telephone conversation by tapping into the termination box, which is usually not secured and is located on the exterior of the telephone customer's premises.

End-to-end confidentiality is more appropriate for heterogeneous, VoIP networks because this approach assumes that the entire end-to-end path is untrustworthy. Additionally, unlike link encryption, it is flexible and scalable. End systems (which are usually maintained by end users) have more flexibility in upgrading to new technology than intermediate network nodes (which are usually maintained by system administrators).

4 Quantifying the Error-Expansion Properties

In this section, we summarize results that quantify the error expansion caused by block ciphers in CBC mode and synchronous stream ciphers. We refer the reader to [5] for the details of our analysis.

4.1 Preliminaries

For the discussion that follows, we assume uncorrelated bit errors as seen at the input to the decryptor. For a typical VoIP system, this is usually a valid assumption because bit errors at this point in the system are residual errors. Corruption enters the system as a result of channel phenomenon (e.g., multipath fading) experienced from any wireless links over the end-to-end path. However, the raw channel errors undergo mitigation techniques such as interleaving and forward error-correction (FEC). If deployed properly, these techniques tend to mask correlation in any residual errors. Even in a slow fading channel, where the fade length might exceed the interleaving depth, other diversity techniques, such as spread spectrum or multiple antennas, can be used to provide the receiver with uncorrelated samples of the signal [11].

For stream encryption, we assume the T^{th} packet contains N bits for cipher synchronization (where $T \in \{1, 2, ...\}$) and packet loss is primarily caused by corrupt packet headers. We consider packet losses from congestion or re-ordering of packets negligible. Although these other loss mechanisms do exist in practice, inclusion of them here confound the analysis. It suffices to say that any increase in packet losses beyond this limitation would further exacerbate error propagation.

4.2 Summary of Results

To quantify the error-expansion properties, we used a three prong approach: 1) we developed stochastic models that describe the error structure of the decryptors for each cipher, 2) using these models, we derived the statistics to describe the length and frequency of error events (defined below), and 3) from these statistics, we derived an expression for the average error expansion.

We define an error event as the state in which the decryptor garbles an integral number of consecutive blocks (or packets) at its output in response to one or more bit (or synchronization) errors at its input. Thus, an error event starts when the decryptor begins garbling recovered plaintext and ends when the decryptor stops garbling recovered plaintext. For block encryption in CBC mode, a corrupt ciphertext block starts an error event, which ends when the decryptor receives the next error-free ciphertext block. For synchronous stream ciphers, a lost ciphertext packet starts an error event, which ends upon re-synchronization. We measure the length of an error event in blocks for block ciphers and packets for stream ciphers.

If we consider the case of Tequals one, then both these error structures can be modeled by a Markov Chain with two states: an error state and an error-free state. This stochastic process is stationary and ergodic, such that the mean time spent in the error state ($\overline{H}$) and mean time spent in the error-free state ($\overline{R}$) can be derived as

$$\overline{H} = \frac{1}{1-\rho} \quad \text{and} \quad \overline{R} = \frac{1}{\rho}\,, \tag{1}$$

where ρ is the transition probability to the error state and $1-\rho$ is the transition probability to the error-free state. The variance of these quantities is given by

$$\sigma_H^2 = \frac{\rho}{(1-\rho)^2} \quad \text{and} \quad \sigma_R^2 = \frac{1-\rho}{\rho^2}\,, \tag{2}$$

respectively. Alternatively, we can interpret $\overline{H}$ as the mean length of an error event and $\overline{R}$ as the mean length between successive error events.

For block ciphers in CBC mode, ρ is the probability the k^{th} ciphertext block c_k is in error. For synchronous stream ciphers, ρ is the probability that in the k^{th} ciphertext packet c_k at least one of the synchronization bits are in error. In both cases, we can express this quantity as

$$\rho = 1 - \left[1 - \overline{BER_{in}}\right]^N\,, \tag{3}$$

where $\overline{BER_{in}}$ is the average, pre-decryption BER and N is the block size in bits (for block ciphers) or N is the number of synchronization bits in each packet (for stream ciphers).

Using these statistics, we can show that the average post-decryption BER ($\overline{BER}_{out}$) is given by

$$\overline{BER}_{out} = \frac{1 - \left[1 - \overline{BER}_{in}\right]^N}{2} . \tag{4}$$

For $\overline{BER}_{in}$ small ($< 10^{-2}$), we can approximate this expression as

$$\overline{BER}_{out} \approx \frac{N}{2} \overline{BER}_{in} . \tag{5}$$

Re-arranging (5), we define the error expansion as

$$\delta \equiv \frac{\overline{BER}_{out}}{\overline{BER}_{in}} = \frac{1 - \left[1 - \overline{BER}_{in}\right]^N}{2\,\overline{BER}_{in}} \approx \frac{N}{2} . \tag{6}$$

Fig. 3 plots (4) over a range of N and $\overline{BER}_{in}$. We observe that $\overline{BER}_{out}$, is directly proportional to N. Additionally, for given N, the error expansion is approximately constant for $\overline{BER}_{in}$ less than 10^{-2}. The two most typical block sizes and number of synchronization bits are N equal to 64 and 128 bits.

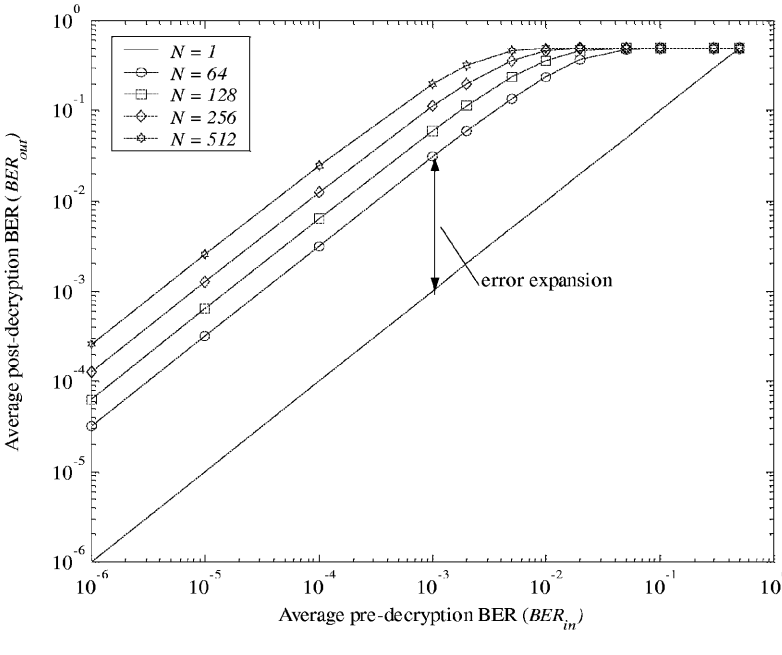

Fig. 3. Comparison of bit error expansion in CBC modes for varying block sizes. The amount of error expansion increases with block size.

Fig. 4 compares the theoretical $\overline{BER_{out}}$ from (4) to an empirical $\overline{BER_{out}}$ measured using DES in CBC mode. The figure shows error expansion of more than an order of magnitude for both cases. We obtained similar results for DES in 64-bit output feedback (OFB) mode, which can be shown to have identical error properties to any synchronous stream cipher that requires 64 synchronization bits [5].

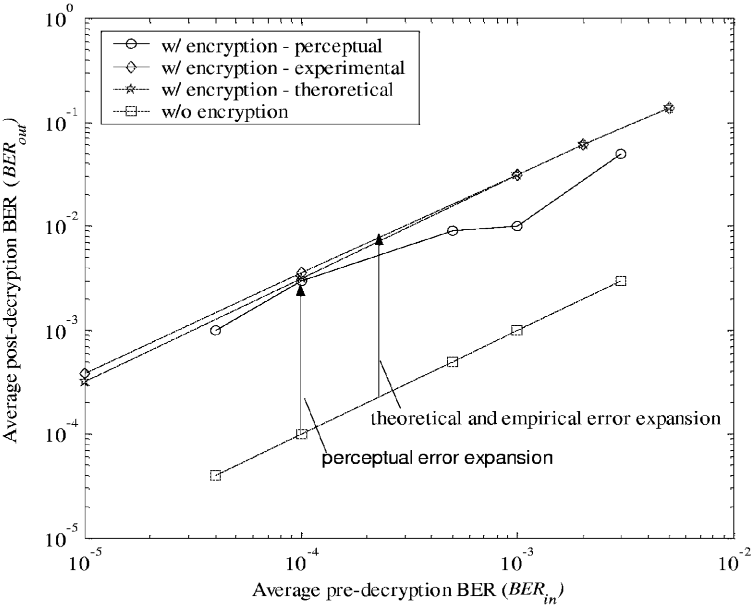

Fig. 4. Comparison of theoretical, experimental and perceptual bit-error expansion for block ciphers in CBC mode.

Fig. 4 also compares the results of a subjective assessment to the theoretical and empirical results. This assessment was conducted using the voice codec of the Global System for Mobile communications (GSM) [10]. The line with circular markers represents a set of twenty users mean opinion regarding their perception of degradation in the average, post-decryption BER. More precisely, the degradation they perceived to maintain the same subjective quality between GSM voice with and without encryption. These results also show error expansion of at least an order of magnitude. Similar results were obtained using 64 kb/s pulse code modulation (PCM).

5 Cost of Confidentiality

To provide good subjective quality for VoIP, it is necessary to define QoS targets for delay, jitter, bit-error rate, and packet-loss rate. We define a QoS target for a given parameter as the minimum (or maximum) value that is acceptable to the VoIP service. Satisfying these targets comes at a cost, which we measure in terms of resources (e.g., memory, processing cycles, power, frequency spectrum, traffic capacity, and channel capacity). For a given QoS target, we show that mitigating error expansion caused by confidentiality leads to increased consumption of resources.

5.1 Reduced Channel Capacity

Forward-error-correction (FEC) coding can mitigate the affect error expansion has on subjective quality. However, for a given modulation scheme, voice codec, and fixed frequency spectrum, applying FEC to compensate for error expansion reduces channel capacity. In particular, enough FEC coding must be applied to compensate for error expansion such that the post-decryption BER satisfies the BER target ($< 10^{-4}$ for VoIP). For a typical scenario (e.g., a 64-bit block cipher), the error expansion caused by confidentiality can be more than an order of magnitude. To improve the BER performance by an order of magnitude, communication systems typically require rate 1/2 (or smaller) FEC codes, especially for high BERs ($> 10^{-3}$). Thus, mitigating error expansion using FEC can reduce the channel capacity by a factor of two or more. In an environment where additional frequency spectrum is scarce, reduction in channel capacity is undesirable, especially for high-bitrate, multimedia applications.

5.2 Increased Delay

Adding any processing in the end-to-end path increases the overall delay. Thus, using FEC coding to mitigate error expansion will increase processing delay. Another way to avoid error expansion is to exclusively use stream encryption over wireless links and block encryption elsewhere. However, this is not an end-to-end approach. This approach requires transcoding at each wired/wireless node to convert from stream encryption to block encryption (and vice versa). Unfortunately, transcoding also increases the end-to-end delay. In delay-sensitive multimedia applications like VoIP, meeting the delay target (which is about 150 milliseconds) is paramount for good QoS. Managing delay for multimedia applications in IP networks is a challenge. Thus, it is desirable to find other means to compensate for error expansion that do not trade-off delay.

Additionally, transcoding can lead to weakened security. At each intermediate node that performs transcoding, the data is temporarily in the clear. Thus, an adversary who is trying to eavesdrop on a multimedia conversation has another potential point of attack. These intermediate nodes might have vulnerabilities, such as swapping secret keys out of main memory to an insecure swap file.

5.3 Reduced Traffic Capacity

In wireless systems, one precious resource is traffic capacity, which is defined as the number of users the system can support simultaneously. In this section, we summarize results that show error expansion can reduce the traffic capacity of a direct-sequence, code-division, multiple-access (DS-CDMA) system by more than an order of magnitude.

We consider a DS-CDMA system that supports variable QoS for VoIP users via the optimal power control algorithm in [12]. The algorithm presented in this paper is optimum in the sense that it minimizes the interference each user experiences from other users (within a given cell), while satisfying each user's reliability requirement. Following this paper, we consider multidimensional 8-PSK, which is one example of combined modulation and coding (with low complexity) that is compatible with the power control algorithm. We consider only two types of users: those with confidentiality and those without. Thus, the power control algorithm provides two distinct levels of reliability: $\overline{BER}_1$ for users without confidentiality and $\overline{BER}_2$ for users with confidentiality. There are a total of M users, $M1$ users without confidentiality and with reliability requirement, $\overline{BER}_1$ and $M2$ users with confidentiality and with reliability requirement, $\overline{BER}_2$. We wish to derive an expression that relates the total system traffic capacity, M, to the fraction of users requiring confidentiality, $M2/M$.

We want the power control algorithm to provide all users with the same subjective quality. Therefore, to compensate for error expansion, we use (6) to obtain

$$\overline{BER}_2 \approx \frac{\overline{BER}_1}{\delta} \,. \tag{7}$$

A feasible solution for this power control algorithm exists and this solution is unique and optimum if and only if

$$\sum_{m=1}^{M} \beta_m \alpha_m < 1 \,, \tag{8}$$

where

$$M = \textit{the total number of users} \,, \tag{9}$$

$$\beta_m = \begin{cases} 1, \textit{ if user } m \textit{ is transmitting} \\ 0, \textit{ otherwise} \end{cases} \,, \tag{10}$$

$$\alpha_m = \frac{(E_b/N_0)_m}{G + (E_b/N_0)_m} \,, \tag{11}$$

$$(E_b/N_0)_m = \textit{the energy per bit to interference for user } m\,, \tag{12}$$

$$G = \textit{spread spectrum processing gain}\,. \tag{13}$$

This feasibility condition is identical for both uplink and downlink cases. Applying (8) to our setup and writing the equation in terms of $M1$ and $M2$ yields

$$M < 1 + \frac{G}{(E_b/N_0)_1} - M2 \left[\frac{\dfrac{(E_b/N_0)_2}{G + (E_b/N_0)_2}}{\dfrac{(E_b/N_0)_1}{G + (E_b/N_0)_1}} - 1 \right] . \tag{14}$$

Equation (14) assumes that G and $(E_b/N_0)_1$ are expressed in absolute units, not in decibels. Since it is often desirable to express parameters in a communication system in decibels, we can show that (14) is equivalent to

$$M < 1 - 10^{[G-(E_b/N_0)_1]/10} - M2 \left[\frac{1 + 10^{[G-(E_b/N_0)_1]/10}}{1 + 10^{\{G-[(E_b/N_0)_1+\Delta]\}/10}} - 1 \right] , \tag{15}$$

where G and $(E_b/N_0)_1$ are expressed in decibel units. The parameter Δ represents the perceptual, error expansion specified in (E_b/N_0). That is, Δ is the additional (E_b/N_0)needed by users with confidentiality such that they can have the same subjective quality as users without confidentiality.

Equation (15) expresses the reliability requirement in terms of the energy-to-interference ratio. For a DS-CDMA system, we can approximate the sum of interference from other users and along multiple propagation paths as being additive white, Gaussian noise (AWGN) [12,13,14]. A fading channel and interference from adjacent cells complicates the analysis, but we can still achieve an error performance that approaches an AWGN approximation with the right techniques to mitigate distortion and provide diversity [11]. We use the AWGN formulation from [15] for 8-PSK modulation to map BER into (E_b/N_0).

Capacity-Percent Confidentiality Curve. Fig. 5 illustrates a capacity-percent confidentiality cost curve for the two cases: high reliability with $\overline{BER}_1$ equal to 6×10^{-5} and low reliability with $\overline{BER}_1$equal to 3×10^{-3}. We can interpret this figure as follows. For a fixed reliability requirement (and therefore fixed subjective quality), the capacity decreases as the fraction of users with confidentiality $(M2/M)$ increases. When all users require confidentiality, the traffic capacity is at its minimum point. In contrast, the traffic capacity is at its maximum point when all users do not require confidentiality. Furthermore, the greater the perceptual error expansion, the faster the drop off in capacity. This follows our mathematical analysis well, since a larger Δ yields a larger multiplicative constant in the second term in (15). With a larger multiplicative constant, the traffic capacity drops off rapidly with more users requiring confidentiality.

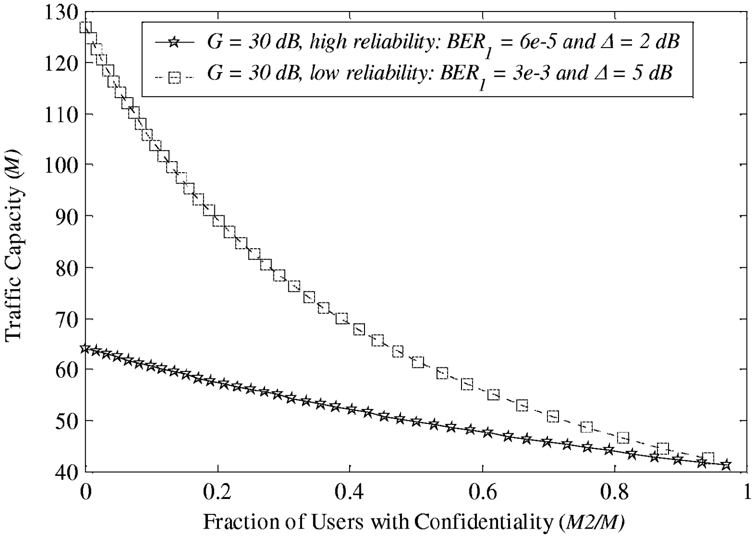

Fig. 5. Traffic capacity-percent confidentiality curve for two cases: 1) high reliability requirement (star markers) and 2) low reliability requirement (square markers).

In addition, the figure shows that the low-reliability curve has a steeper drop-off than the high-reliability curve. This results from the fact that, when all users do not require confidentiality, the system can support almost twice as many users at the lower reliability. Thus, there is more flexibility in resource utilization and admissions control when users require low reliability.

6 Error-Robust Encryption for Voice over IP

To eliminate this trade-off between confidentiality and QoS, we propose error-robust encryption. In this section, we introduce one such method called the automatic synchronization protocol (ASP), which is a technique that makes synchronous stream ciphers robust to synchronization errors.

6.1 Design Goal and Philosophy

Our goal is to design a synchronization technique for synchronous stream ciphers suitable for VoIP over wireless. Our approach to designing ASP is to use existing cryptographic mechanisms as much as possible, while achieving our design goal and satisfying the following properties:

1. Packets should not contain any additional bits specifically for cipher synchronization.

2. The technique should be memory less. That is, correct synchronization of the i^{th} packet should depend only on the integrity of the i^{th} packet's header and not on previous (or future) packets.
3. The security strength of the cipher should be no less than the strength of the underlying cryptographic mechanisms.
4. The design should be flexible enough to be used with any synchronous stream cipher or block cipher in OFB mode.
5. The design should be efficient.

6.2 Description

For each received ciphertext packet, ASP re-initializes the cipher to a different starting point by seeding it with a new key. Fig. 6 depicts how ASP maintains synchronization for synchronous stream ciphers. For each received ciphertext packet, the ASP module produces a d-bit secret key (K_i). Each K_i then re-initializes the keystream generator to a new starting point. ASP takes as input an s-bit sequence number (S_i) that it extracts from each packet header, an n-bit random initialization vector (IV), and a k-bit secret key ($K1$). Both IV and $K1$ are exchanged during session establishment.

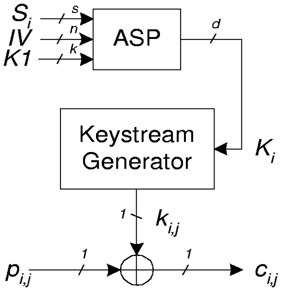

Fig. 6. ASP operation with a generic synchronous stream cipher. S_i is a s-bit sequence number for the i^{th} packet. IV is a n-bit random initialization vector. $K1$ is a k-bit secret key for session. K_i is a d-bit secret key for the i^{th} packet. $k_{i,j}$ is the j^{th} keystream bit for the i^{th} packet. $p_{i,j}$ is the j^{th} plaintext bit for the i^{th} packet. $c_{i,j}$ is the j^{th} ciphertext bit for the i^{th} packet.

ASP is a pseudo random number generator (PRNG) that produces a new secret key for each packet. It is based on existing cryptographic mechanisms, namely block ciphers in counter mode and hash functions. Using PRNGs for generating secret keys and IVs is common practice. The literature contains numerous PRNG methods that use existing cryptographic mechanisms [17,18,19, 20,21]. However, all the methods we reviewed were incompatible with our design goal because they were designed for reliable transport. That is, they were designed with the intent that the sender would generate the pseudo random output, then reliably communicate that output to the receiver. This approach works fine during session establishment, but is inadequate for generating a pseudo random output to be used with each packet in a continuous stream of packets. Yarrow [17], discloses a well designed PRNG from a security perspective, which could be adapted to be compatible with our target applications. However, such a modified Yarrow PRNG would require additional state to maintain its own internal synchronization. Since our goal is to provide cipher synchronization for our target applications, we thought it best to devise a new method that is optimized in this sense.

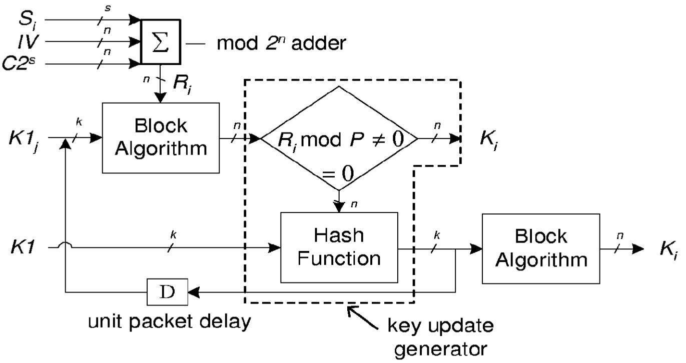

Fig. 7. Internal structure of ASP. S_i is a s-bit sequence number for the i^{th} packet. IV is a n-bit random initialization vector for the session. $K1$ is a n-bit secret key for the session. $K1_j$ is a n-bit secret key for packets between j^{th} and $(j+1)^{th}$ update. $K1_0 = K1$. C is a sequence number overflow counter.

Fig. 7 illustrates the internal structure of ASP. There are three major components: 1) an n-bit block encryption algorithm with a k-bit key, 2) a modulo 2^n adder, and 3) a hash function. The block algorithm is configured in counter mode, where its input is the output of the adder block. The adder block is essential to the technique. Its inputs are chosen in such a way as to transform a rather small sequence number into a much larger n-bit index (R_i). This is done so that the pseudo random output of the block algorithm will have a long period (i.e., 2^n instead of 2^s). The hash function is used every P packets to update the j^{th} session key, $K1_j$. This key-update generator protects ASP output against key compromises. If an adversary compromises the j^{th} session key (for $j > 1$), this adversary will only recover data between the j^{th} and the $(j+1)^{th}$ session keys.

6.3 Voice over IP Example

We implemented ASP in a VoIP application called Speak Freely [22]. Using simulated errors, PCM voice coding, and DES encryption in 64-OFB mode, we performed computer-to-computer VoIP communications over the Internet for three test cases: 1) no encryption, 2) 64-OFB encryption with 128-ASP and 3) 64-OFB encryption with 64 synchronization bits per packet. In Fig. 8, we plot the results of this experiment for the average, post-decryption BER ($\overline{BER_{out}}$) versus the average pre-decryption BER ($\overline{BER_{in}}$). The results for the test case with 128-ASP was indistinguishable from the test case without encryption. That is, there was zero empirical or perceptual error expansion with ASP. In contrast, the third test case exhibits more than an order of magnitude error expansion.

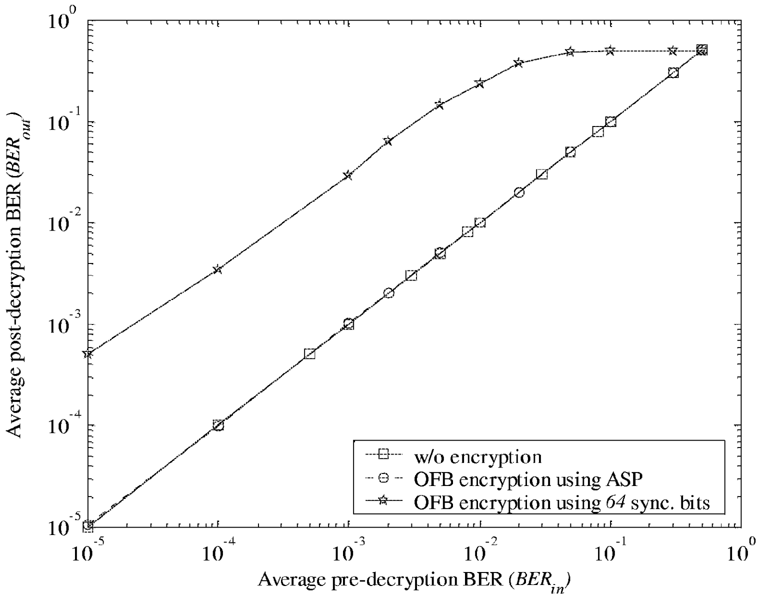

Fig. 8. The average post-decryption BER for three test cases: 1) no encryption (square markers), 2) 64-OFB DES encryption with 128-ASP (circle markers) and 3) 64-OFB DES encryption with 64 synchronization bits per packet (star makers).

7 Conclusions

In this article, we showed how the error properties of secret-key ciphers leads to error expansion for real-time, multimedia applications, particularly in heterogeneous networks. For typical secret-key ciphers, this error expansion can be more than an order of magnitude. For this class of applications, we showed that traditional techniques to combat corruption are inadequate because they consume precious resources. Alternatively, we proposed error-robust encryption and introduced one technique called the automatic synchronization protocol (ASP). This technique is a flexible synchronization means that makes synchronous stream ciphers robust to errors. ASP was designed to work with any synchronous stream cipher or block ciphers configured in OFB mode. We demonstrated its utility for computer-to-computer VoIP communications, but it should also be applicable to digital IP telephones. We hope this article will encourage the wireless networking and multimedia applications communities to view security at the system level, from the perspective of good QoS as well as strong security.

References

1. A. Webster and S. E. Tavares, On the design of S-boxes, in H. C. Williams (Ed.): Advances in Cryptology - Proc. of CRYPTO '85, Springer-Verlag, NY, pp. 523-534, 1986.
2. National Institute for Standards and Technology, Data Encryption Standard (DES), FIPS PUB 46-2, US Department of Commerce, Dec. 30, 1993.
3. X. Lai, On the design and security of block ciphers, ETH Series in Info. Proc., vol. 1, Konstanz: Hartung-Gorre Verlag, 1992.
4. National Institute for Standards and Technology, DES Modes of Operation, FIPS PUB 46-2, US Department of Commerce, Dec. 2, 1980.
5. J. Reason, *End-to-end Confidentiality for Continuous-media Applications in Wireless Systems*, Dissertation, UC Berkeley, December 2000.
6. S. Casner, R. Frederick, and V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, IETF, RFC 1889, January, 1996.
7. C. Bormann et al, RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed, IETF, draft-ietf-rohc-rtp-09, Feb. 2001.
8. Bo Li; Hamdi, M.; Dongyi Iang; Xi-Ren Cao; Hou, Y.T., QoS enabled voice support in the next generation Internet: issues, existing approaches and challenges, IEEE Communications Magazine, Volume: 38 Issue: 4, April 2000
9. The Bluetooth Special Interest Group, Bluetooth Specification, http://www.bluetooth.com, Dec. 1999.
10. Recommendation GSM 06.10, GSM Full Rate Speech Transcoding, ETSI, Feb. 1992.
11. B. Skylar, Rayleigh fading channels in mobile digital communication systems part ii: mitigation, IEEE Communications Magazine, pp. 102-109, July 1997.
12. L. Yun and D.G. Messerschmitt, Power control and coding for variable QOS on a CDMA channel, Proc. IEEE MILCOM Conf., Fort Monmouth, NJ, Oct 2-4, 1994.
13. R. L. Pickholtz, et. al., Spread spectrum for mobile communications, IEEE Transactions on Vehicular Technology, Vol. 40, NO. 2, pp. 313-321, May, 1991.
14. M. B. Pursley, Performance evaluation for phase-coded spread-spectrum multiple-access communication-part I: system analysis, IEEE Trans. on Comm., Vol.COM-25, NO. 8, pp. 795-799, August 1977.

15. P. J. Lee, Computation of the bit error rate of coherent M-ary PSK with Gray Code bit mapping, IEEE Trans. on Comm., Vol. COM-34, NO. 5, pp. 488-491, May, 1986.
16. B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, The Twofish Encryption Algorithm, John Wiley & Sons, 1999.
17. J. Kelsey, B. Schneier, and N. Ferguson, Yarrow-160: Notes on the Design and Analysis of the Yarrow Cryptographic Pseudorandom Number Generator, Sixth Annual Workshop on Selected Areas in Cryptography, Springer Verlag, August 1999.
18. ANSI X9.17 (Revised), American national standard for financial institution Key management (wholesale), American Bankers Association, 1985.
19. P. Gutmann, Software generation of random numbers for cryptographic purposes, Proceedings of the 1998 Usenix Security Symposium, 1998.
20. National Institute for Standards and Technology, Key Management Using X9.17, FIPS PUB 171, US Department of Commerce, 1992.
21. P. Zimmermann, The Official PGP User's Guide, MIT Press, 1995.
22. J. Walker and B. C. Wiles, Speak Freely, http://www.speakfreely.org, 1999.