A SELF-SYNCHRONIZING CASCADED CIPHER SYSTEM WITH DYNAMIC CONTROL OF ERROR PROPAGATION

Norman Proctor
SYTEK, Incorporated
1225 Charleston Road
Mt. View, California 94039-7225

ABSTRACT

A cipher system used for secure communication over a noisy channel can automatically synchronize the sender and receiver by computing a stateless function of a key and a limited amount of the recent ciphertext. The more ciphertext feedback is used, the more the errors from the noisy channel are propagated. The less feedback is used, the easier ciphertext-only and chosen-plaintext attacks become. There is a trade-off between security and noise that must be made when a self-synchronizing system is built.

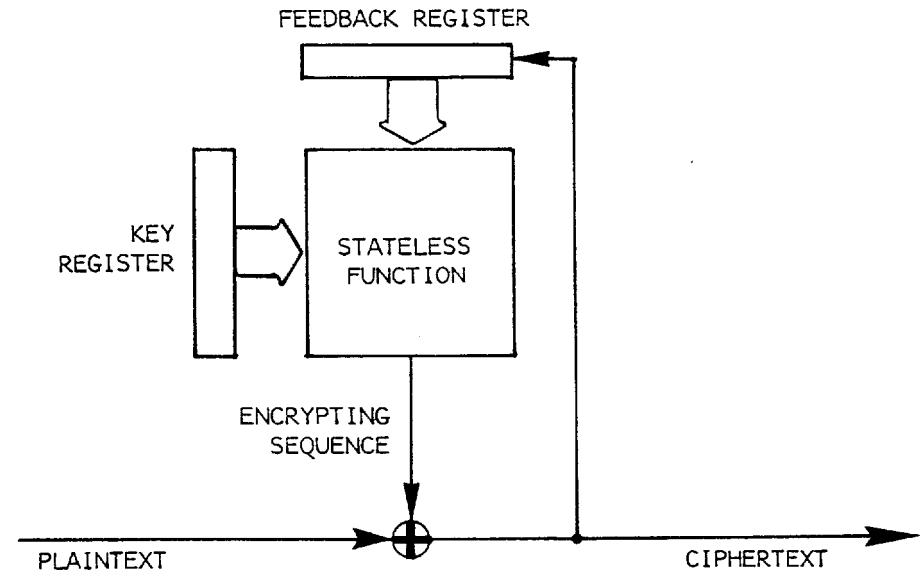
This paper presents a self-synchronizing cascaded cipher system that permits most combinations of key and ciphertext feedback lengths and also allows adjustment of the trade-off between security and noise during system operation. At times when maximum security is not needed, the error propagation can be reduced temporarily.

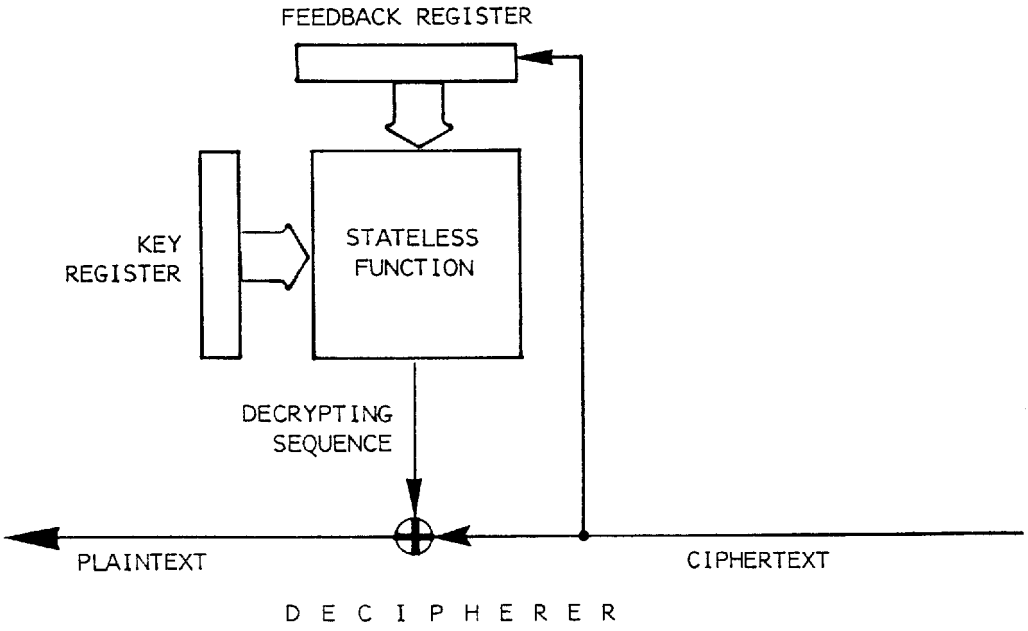
As implemented in hardware, the cascaded cipher has a storage register for each stage. The function computed would normally depend on the state of this storage, but different clocks are used at each stage to render the function stateless. The use of a cascade helps to keep the hardware cost down.

CIPHER FEEDBACK SYSTEMS

Cipher feedback systems are useful in circumstances where synchroization between the encipherer and decipherer cannot be guaranteed in any practical way. Figure 1 shows a generic cipher feedback system.

G.R. Blakley and D. Chaum (Eds.): Advances in Cryptology - CRYPTO '84, LNCS 196, pp. 174-190, 1985.





CIPHER FEEDBACK SYSTEM

FIGURE 1

The encipherer consists of a k-bit key register, an f-bit feedback shift register and a combinatorial circuit with inputs from the two registers and with one output. The sequence of output bits will be called the encrypting sequence. It is a pseudo-random sequence if the combinatorial circuit is correctly chosen. Some constraints may need to be placed on the choice of keys in order to achieve pseudo-randomness. The encrypting sequence is combined with the plaintext sequence by modulo-2 addition to produce the ciphertext sequence. As each bit of the ciphertext sequence is produced, it is shifted into the feedback register. This means that each ciphertext bit may depend on the k bits of the key, the previous f bits of the ciphertext and the current plaintext bit.

The decipherer like the encipherer has a k-bit key register, an f-bit feedback register and a combinatorial circuit identical to the one in the encipherer. The decrypting sequence from that circuit is combined with the ciphertext sequence to produce a plaintext sequence. After each bit of the ciphertext sequence is first used, it is shifted into the feedback register so that it is also used to produce the next f bits of the decrypting sequence.

SELF-SYNCHRONIZATION

A bit of the decrypting sequence will be the same as the corresponding bit of the encrypting sequence provided only that the encipherer and decipherer are using the same key and that the previous f bits of ciphertext received by the decipherer are unchanged since the encipherer produced them. We will assume in general that some technique ensures that the encipherer and decipherer do use the same key. The second proviso will not always be met if the channel that carries the ciphertext is noisy for any reason.

Any difference between ciphertext bits produced by the encipherer and the corresponding bits received by the decipherer will be called errors in the ciphertext. Similarly, differences between corresponding bits in the two plaintext sequences or in the encrypting and decrypting sequences will be called errors in the plaintext or in the decrypting sequence.

A bit of the decipherer's plaintext sequence must be the same as the corresponding bit of the encipherer's plaintext sequence provided only that the corresponding encrypting and decrypting sequence bits are the same and the current ciphertext bit received is the same as the current ciphertext bit produced. This means that following a transmis-

sion error which changes some bits in the ciphertext, there can be errors in the plaintext for at most f bits after the last erroneous ciphertext bit was received. Because the decipherer recovers automatically within a short and fixed number of bits to errors in the ciphertext, cipher feedback systems are said to be self-synchronizing.

Self-synchronization makes it unnecessary for the encipherer and decipherer to agree on a value or time to initialize their feedback registers. The first f bits of the plaintext sequence are not part of the message because the decipherer cannot be expected to recover them. But barring errors in the ciphertext, the decipherer can recover all the rest of the plaintext sequence no matter what its feedback register held when the first ciphertext bit was received.

Self-synchronization also makes it feasible for the decipherer to interpret the ciphertext stream as a sequence of bits by periodic sampling even though its clock is not exactly as fast as the encipherer's clock. A slight difference in clock speeds would produce occasional bursts of plaintext errors up to f + 1 bits long but would not produce complete gibberish.

There are variants on the system in Figure 1 which retain the self-synchronizing property. The encipherer could use a different invertible computation to produce the ciphertext sequence from the encrypting sequence and the plaintext sequence instead of the self-inverting modulo-2 addition. The decipherer would use the inverse. Another variation has the plaintext and ciphertext processed block by block instead of bit by bit with each ciphertext block (or part of it) being shifted into the feedback registers. We will not explicitly consider such variants, but the discussions below would be analogous or even identical for the variant cipher feedback systems.

The essential features are that (1) the encipherer's and decipherer's feedback registers contain a limited number of bits of the recent ciphertext sequence (or equivalent information), (2) the encrypting and decrypting sequences are produced by the same stateless function of the key and at least some of the contents of the feedback register, and (3) the encrypting sequence is combined with the plaintext in such a way that the plaintext can be recovered from the resulting ciphertext whenever corresponding portions of the encrypting and decrypting sequences match.

THE SECURITY PROBLEM

The advantages of self-synchronization come with two major disad-

vantages. One is that errors in the ciphertext are propagated. The other is that the length of the feedback register may allow the cipher system to be broken by an exhaustive search less expensive than an exhaustive search of the key space.

As long as the key remains the same, the bit of the encrypting sequence produced after a particular f-gram in the ciphertext will always be the same. A codebook can be assembled whose index is the values of the previous f bits of the ciphertext and whose entries are the bits of the encrypting sequence that must follow. It can be used in place of the key and combinatorial circuit in the encipherer or decipherer. The codebook has 2^f bits. If 2^f is less than the key length k, it is easier to search exhaustively for the right codebook than to search for the right key.

When the codebook has fewer bits than the key, many keys must be equivalent to each other. Each equivalence set of keys is associated with a unique codebook. The mean size of the equivalence sets will be at least 2 raised to $(k - 2^f)$. Analysis of the plaintext and ciphertext can identify the key's equivalence set by finding that the set's codebook (or some member of the set) deciphers the ciphertext. But unless the equivalence set turns out to be a singleton set, the key cannot be positively identified from the plaintext and ciphertext. Identifying all the members of the key's equivalence set is more difficult than finding the set's codebook. This fact is useful as we will discuss later if the secrecy of the key can sometimes be more important than the secrecy of the plaintext.

THE NOISE PROBLEM

A feedback register which is too short relative to the key register can make the cipher feedback system easier to break than it would otherwise be. That makes very short feedback registers undesirable. The error propagation from cipher feedback registers, on the other hand, makes very short feedback registers attractive.

In most other cipher systems, ciphertext errors either are not propagated at all or are potentially propagated indefinitely. If only error-free plaintext were acceptable, indefinite propagation would be preferable as it makes error detection easier. Instead, we will assume that some errors in the plaintext are tolerable but that it is valuable to minimize them. Clearly, the best error propagation is none at all. The error rate in the plaintext would then be no worse than the rate in the ciphertext.

With cipher feedback systems, an error in one bit of the ciphertext is normally propagated within the next f bits of the plaintext. This is because the decipherer's feedback register holds the erroneous bit while the next f bits of the decrypting sequence are produced. Any of those f bits in the decrypting sequence could be in error as a result. Unless it is cancelled by another error in the ciphertext, an error in the decrypting sequence means an error in the plaintext.

One plaintext error is unavoidable for each ciphertext error, but with error propagation there will be extra plaintext errors. The ratio of the expected number of extra plaintext errors to the expected number of ciphertext errors will be called the increased noise N. The increased noise may depend on the key value, the f ciphertext bits that precede an erroneous ciphertext bit, the f ciphertext bits that follow it and the combinatorial circuit that computes the decrypting sequence. To examine the relationship between N and f, we will assume that in order to achieve pseudo-randomness in the encrypting sequence, a combinatorial circuit was chosen for which changing a non-empty subset of the bits in the feedback register has a fifty percent probability of changing the circuit's output. This probability is based on a uniform distribution of original values for the feedback register. A uniform distribution is expected when the encrypting sequence is pseudo-random. The probability is assumed to hold for all keys or at least for all keys that are ever used.

If there were no error propagation, N would be zero. If all ciphertext errors were single bit errors and were separated by at least f correct bits, N would be f/2. If the ciphertext errors were not always so well separated, they could be thought of as bursts. We will consider two errors separated by fewer than f correct bits to be in the same error burst. Bursts are thus separated by at least f correct bits but contain no stretches of correct bits that long.

If all ciphertext errors were in bursts of at least two bits, N would be $(f + 1)/2pb + (1 - 2p)/2p$ where b is the mean length of the bursts and p is the probability that a bit within a burst is in error. The first and last bits of a burst must be in error to define the burst's position and length, and we assume that the intervening bits each have the same independent probability of being in error. That probability is $(bp - 2)/(b - 2)$ and can be expected to fall between zero and one half.

If single bit errors are mixed with the burst errors, N rises toward f/2.

The increased noise is roughly proportional to the length of the

feedback register if the assumptions about the combinatorial circuit are met or nearly met. If as is likely to be the case, keeping the ciphertext error rate low is expensive and low plaintext error rates are valuable, only very short feedback registers may be practical.

USING LESS FEEDBACK

Deciding on a length for the feedback register is a problematic issue in selecting a cipher feedback system for an application. If the length is too short relative to the key length, the security of the system may be lost. If it is too long relative to the expected error bursts in the channel carrying the ciphertext, the plaintext may be too noisy to be useful or the cost of improving the ciphertext channel may be exorbitant.

One way to deal with the dilemma is to postpone it. It is better to choose a length for the feedback register for each message to suit the message's sensitivity and the condition of the ciphertext channel when the message is sent than it is to choose the length based on the worst case messages.

Some messages require the full measure of security that the key length provides. They should be sent when the channel noise is relatively quiet so that one can use a feedback register long enough not to impair security and still not cause an intolerable error rate in the plaintext produced by the decipherer.

Other messages may require some cryptographic protection but not as much. Not all secret messages need absolute protection. A message whose secrecy can be priced is protected for all practical purposes if the cost of thwarting the protection exceeds the value of the message's secrecy. Such messages can be sent even when the channel noise would be too bad for the long register by using only some of the register's length.

If only part of the feedback register is used as input to the stateless function that computes the encrypting and decrypting sequences of a feedback cipher system, the size of the codebook depends on the length used. For example, if the feedback register has ten bits of which only five are used to compute the encrypting and decrypting sequences, the codebook has thirty-two bits not a kilobit. The cost of breaking the system would be the cost of trying 2^{32} codebooks assuming the key has at least thirty-two bits and the system has no other weaknesses. For some messages in some applications, this may be adequate

protection.

The effect on the noise of only using part of the feedback register depends on which part is used and on the nature of the errors in the ciphertext. If nearly all errors in the ciphertext are single bit errors, the increased noise in the plaintext depends on the length of the part used. For the example above in which only five bits of the feedback register are used, the increased noise would be the same as for a five-bit register all of which was used.

On the other hand, with bursty ciphertext errors, the increased noise depends primarily on the position of the bit used which is furthest from the input to the feedback register. For the example above in which five out of ten bits are used, the feedback register's bits are numbered from one to ten starting with the bit nearest the input. If the five bits used are the even-numbered bits, the increased noise would be as bad as if all ten bits were used. The noise would be just as bad if bits six through ten were used. But if bits one through five are the ones used, the increased noise is the same as if the feedback register only had five bits. In general, the greatest benefit in noise control comes from prefering to use the bits closest to the input.

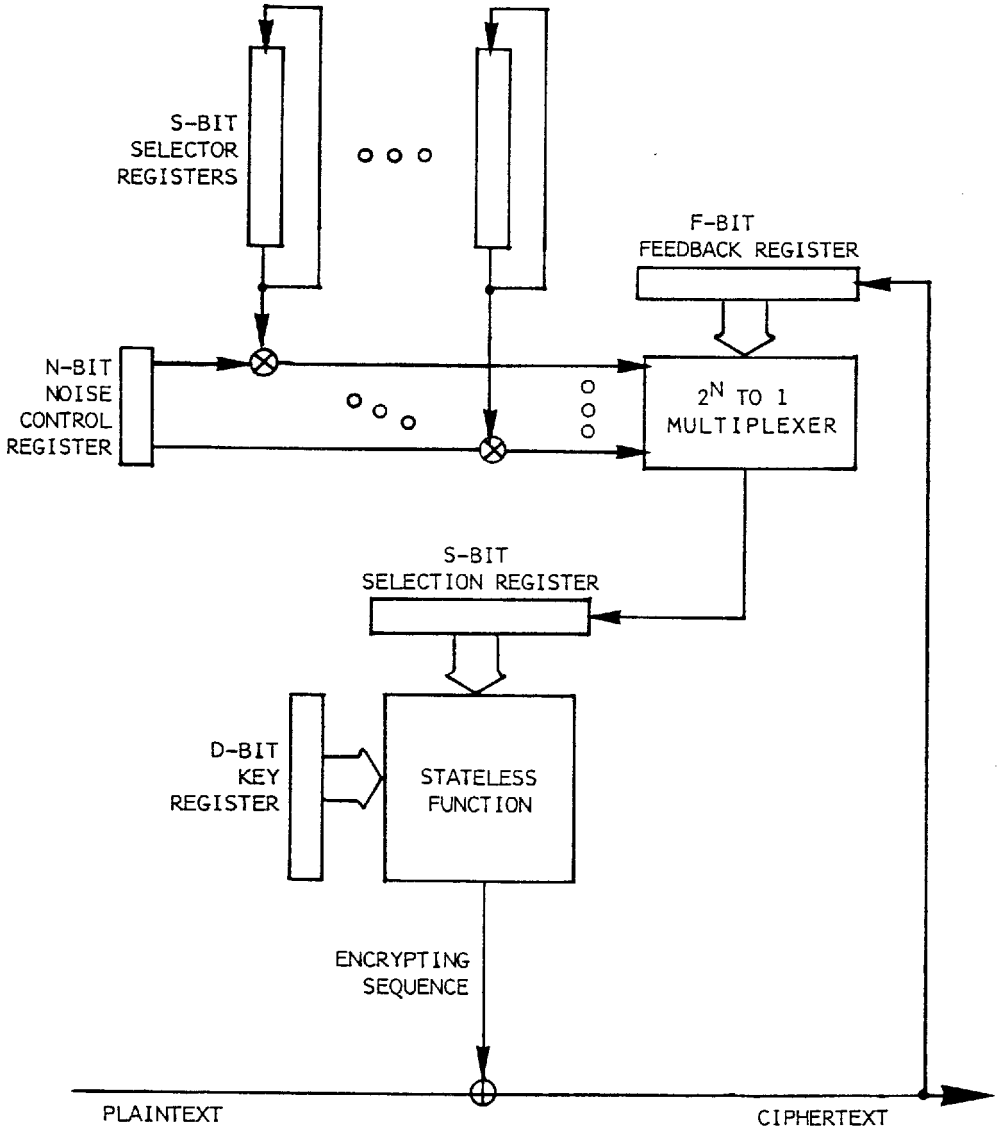
A SOLUTION

Figure 2 shows the encipherer of a cipher feedback system allowing a dynamic choice of the effective feedback length. It has an f-bit cipher feedback register, an n-bit noise control register, a d-bit key register, an s-bit selection register, and n s-bit circulating selector registers. All the registers are shift registers except possibly the noise control register and the d-bit key register.

The encipherer has a 2^n-to-1 multiplexer with data inputs from the feedback register and address inputs from n modulo-2 multipliers. The multipliers each take an input from the noise control register and an input from one of the selector registers. The output from the multiplexer is input to the selection register.

The encipherer also has a stateless function with inputs from the selection register and the d-bit key register. The function's output is the encrypting sequence.

The contents of the n selector registers, the d-bit key register and the noise control register constitute the key. That makes the key length k be ns + d + n. Later discussion will show that k may actually be only ns + d since the noise control register is not like the other key registers.



ENCIPHERER WITH DYNAMIC NOISE CONTROL

FIGURE 2

The multiplexer's n address inputs and f data inputs make n be the base two logarithm of f, or the round-up of that log if f is not a power of two. For now we will assume that the length of the feedback register is a power of two.

The constant s should be at least the base two log of d. It is mostly constrained by the options for clock speeds.

There are two clocks called the feedback clock and the selection clock. The feedback clock shifts the feedback register. It pulses once for each ciphertext bit, or from the other points of view, once for each plaintext bit or once for each bit of the encrypting sequence.

The selection clock pulses s times faster than the feedback clock. It shifts the selector registers and the selection register. Thus for each bit of the encrypting sequence, the selector registers circulate exactly once and the selection register is filled with entirely new multiplexer outputs. This means that the encrypting sequence is a stateless function of the key and the feedback despite the existence of a selection register and the shifting of the selector registers. The noise control register and the d-bit key register remain fixed.

Each multiplexer output is called a feedback selection because it is a bit from the feedback. The products which are the outputs from the n multipliers determine which feedback bit is selected for each selection. The rightmost bit of the feedback register is selected when all the products are zeros. And when they are all ones, it is the leftmost bit. The other combinations of products select the other bits of the register according to the binary number formed by the products.

A zero in a bit of the noise control register leaves only half of the feedback bits able to be selected because the output from the multiplier receiving that zero will itself be zero regardless of the input from the selector register. In fact, a zero in the noise control register causes one of the selector registers to have no influence on any selections. Two zeros allow only one quarter of the feedback bits to be selected. Each additional zero eliminates half of the remaining choices. Having all zeros in the noise control register allows only the rightmost bit of the feedback register to be selected.

Because of the addressing order of the feedback bits, the bits that can be selected are the rightmost bits when the zeros in the noise control register are the bottommost bits. For example, if the noise control register has only two ones and they are the two topmost bits of the register, the s selections that go into the selection register will each be one of the four rightmost bits of the feedback register. Which of the four is selected in each case is determined by the corresponding bits in the two leftmost selector registers.

As we noted earlier, prefering the rightmost feedback bits, the ones nearest the input to the register, causes the least additional noise when ciphertext errors are bursty. (When errors are not bursty, preferences do not matter.) In the example, the noise would drop to the level expected of a feedback register with only four bits.

The decipherer for the cipher feedback system produces its decrypting sequence exactly as the encipherer produces the encrypting sequence. To recover the plaintext, the decipherer must be using the same key as the encipherer. The contents of the noise control registers must be the same, the contents of the d-bit key registers must be the same, and the contents of the corresponding selector registers must be the same when they are at the same point in their circulations. For each selector register, it is sufficient to know that the encipherer's register when its feedback clock pulses matches the decipherer's register when the decipherer's feedback clock pulses. It does not matter whether the two feedback clocks pulse at the same time.

OTHER FEEDBACK REGISTER LENGTHS

We had assumed that the length of the feedback register f was a power of two. The selection process is more complicated otherwise because some feedback bits can be selected by more than one address input to the multiplexer. This makes for irregularities in the effects of zeroes in the noise control register. As an example, we will consider a cipher feedback system in which f is 11. That makes n be 4. We will number the feedback bits from right to left as 1 through 11. The address input to the multiplexer will be referred to as the decimal equivalent of the binary number formed by the products with the topmost product being the most significant binary digit. The noise control value is similarly derived from the noise control register's contents with the topmost bit being most significant. The multiplexer selects bits from the feedback register according to the following table:

Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Selection	1	2	3	3	4	5	5	4	6	7	7	8	9	9	10	11

The next table shows the number of feedback bits that can be selected for each noise control value. For example, a noise control value of eleven (1011) limits addresses to 0, 1, 2, 3, 8, 9, 10 and 11. This in turn allows selections to be made only from six feedback bits, bits 1, 2, 3, 6, 7, and 8.

Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Selectable	1	2	2	3	2	4	4	5	2	4	4	6	4	7	8	11

Provided that errors are not bursty, an appropriate choice of the noise control register's contents allows this feedback cipher system to be used with the same noise level as expected from other feedback cipher systems whose feedback registers had any number of bits up to eight. It could also be used with the noise level expected of its actual length of eleven.

Assuming that the key length is between 64 and 128, the noise control values of thirteen or more would not reduce the security of the system and the other noise control values would provide six different levels of security. Some of those six levels are probably too low for any messages in the application, but some choice still remains if not all messages require the security of having to try more than 2^{64} codebooks or keys.

When errors are bursty, the noise level depends primarily on the distance from the input to the feedback register of the furthest bit that can be selected. The following table shows what that distance is for each noise control value. For example, the furthest bit that can be selected when the value is eleven is bit 8.

Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Distance	1	2	3	3	4	5	5	5	6	7	7	8	9	9	10	11

Because distances from one through eleven all appear in the table, the system can have the noise level expected of a cipher feedback system whose feedback register's length was any number up to eleven. Assuming again that the key length k is between 64 and 128, the choices that might be attractive are error control values of 0, 1, 3, 7, 11 and 13 which require trying 4, 16, 256, 2^{32}, 2^{64}, and 2^{k} codebooks or keys, respectively, and give effective feedback lengths of 1, 2, 3, 5, 8 and 9, respectively.

DYNAMIC NOISE CONTROL

The contents of the noise control register have been considered as part of the key. This is appropriate from some points of view but not from other points of view. If an attacker can measure the error rate, it is best not to consider the noise control value as part of the key. The error rates reveal too much about the noise control val-

ue to call it secret. On the other hand, the encipherer and decipherer must agree on the contents of their noise control registers just as with other registers that hold key material. If one changed the noise control value but not the rest of the key, the other would have to do the same.

There is another reason for the noise control value not to be secret like the key. It may be desirable to change the value much more often or more impulsively than the key is changed. If so, it would be convenient to communicate the new noise control values by an insecure channel.

Changing just the noise control register is not always safe. As an example of the danger, consider a cipher feedback system in which f is 8, n is 3, d is 30, and s is 10. k is 60, not including the noise control value in the key. Some messages are worth less than the cost of trying 64K codebooks, but others are worth more than the cost of trying 2^{50} keys. If the cheaper messages were sent when there was one zero in the noise control register, a set of all possible values for the d-bit key register and two of the three selector registers could be found by trying 2^{50} keys with the same value in the third selector register. The set is expected to have 2^{34} members. This exposes the cheaper messages but at too high a price if they were the only target. Then when the zero is changed to a one and the key is not changed, the full value of the key can be found by trying 2^{44} keys. A change of key would have been adequate to protect the more valuable messages, but changing just the noise control value was not enough.

Simple modifications to the registers holding key material can make it safe to change the noise control value without changing the key. One option is to eliminate the d-bit key register and use various positions from the n circulating selector registers instead for the inputs to the stateless function in Figure 2. The key length k would then be ns. Another option is to have only one circulating selector register and use n different positions for the inputs to the multipliers. This makes k be s + d. Combining those options reduces k to s.

Any of these options requires s or d to be larger to keep the same key length k. Hardware constraints are likely to put an upper limit on s and might sometimes limit k too much for some of these choices to be viable.

With any of the options, using a key with a short effective feedback length does not compromise the prior or subsequent use of the same key with a longer effective feedback length. Returning to the same example, while the cheaper messages were being sent with an effective feedback length of four bits, the codebook could be found by trying 64K

codebooks. This costs more than the cheaper messages are worth and is no help in exposing more valuable messages. Trying all 2^{60} keys is expected to give a set of 2^{44} possible keys, but trying fewer keys would give only an incomplete set of possible keys. After changing to an effective feedback length of eight for the more valuable messages, it would be worth searching the set of keys to expose those messages but the cost of producing the set exceeds the value of the more valuable messages.

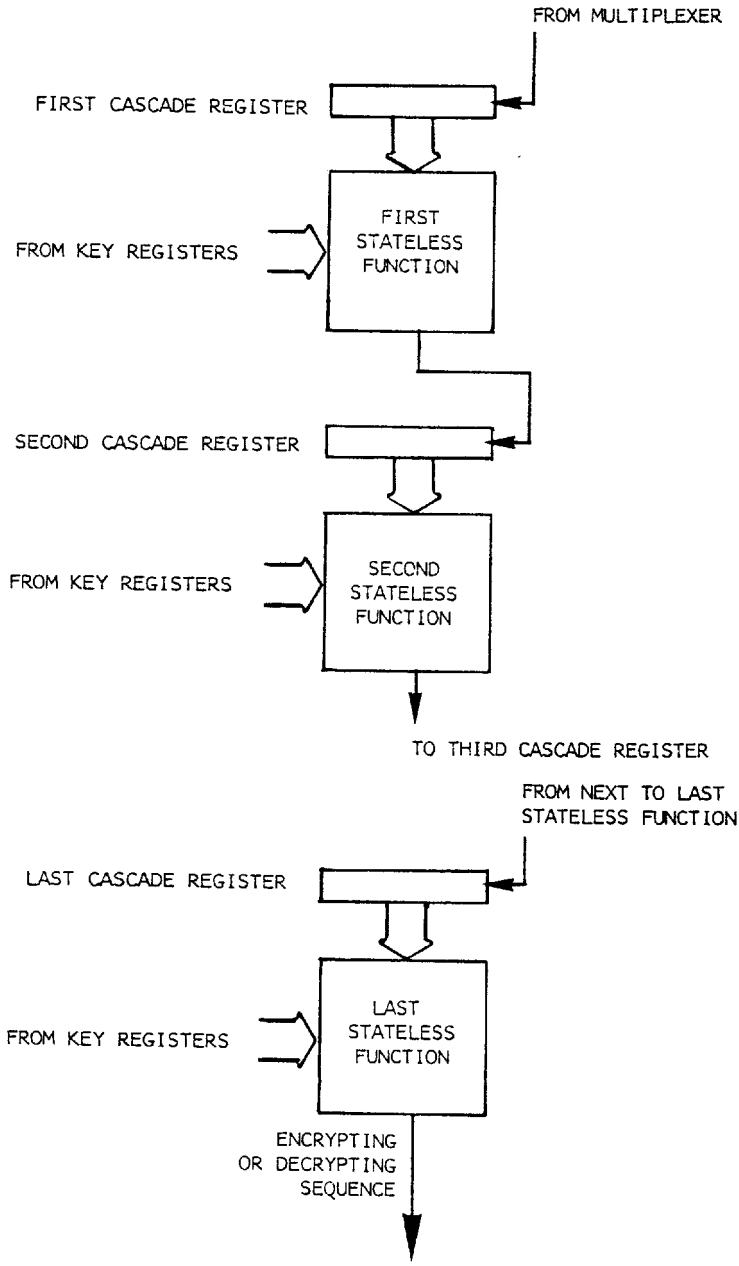
Besides having two or more classes of messages with different values, having a choice of security levels and effective feedback lengths can be worthwhile if individual messages are not very valuable but the value of large aggregations of messages substantially exceeds the sum of their individual values. In an application with such messages, the noise control value can be adjusted to provide the maximum security at any particular time that does not produce intolerable noise in the plaintext. When the ciphertext channel is especially noisy, the security is too low for large aggregates but high enough for individual messages. Most of the time the ciphertext channel is not that noisy and the security level is high enough to protect the large aggregates of messages. Enough of the messages get the better protection that the valuable aggregates cannot be compiled at an affordable cost. The benefit is that messages do not need to be postponed when the ciphertext channel is especially noisy and that the ciphertext channel does not need to be as quiet all the time as the higher level of security and tolerance for plaintext noise would otherwise require.

CASCADED SYSTEMS

The encipherer in Figure 2 has an s-bit selection register and a stateless function with inputs from each of those s bits. The stateless function must be non-linear with respect to the selections if the whole system is to be non-linear. Also, s should be about as large as f or larger than f. A substantial portion of the cost of the system is the cost of the selection register and the stateless function.

Figure 3 shows an alternative to the selection register and stateless function of Figure 2. The original stateless function is replaced by a cascade of stateless functions. The total cost of the functions in each stage of the cascade is likely to be much less than the cost of the one large function. The key inputs to each stateless function come from the key registers.

Another savings comes from the replacement of the selection register with a much shorter cascade register for each stage in the cascade.



CASCADE

FIGURE 3

We will call the length of the cascade register for the i'th stage c_i. The product of the lengths of the cascade registers is s, the length of the circulating selector registers.

The selections output from the multiplexer are input to the first cascade register. The outputs from each stateless function are input to the cascade register of the next stage in the cascade. The last stage is an exception. Its output is the encrypting (or decrypting) sequence.

Each stage in the cascade uses a different clock for shifting its cascade register. The first stage uses the selection clock, the clock which shifts the circulating selector registers. The clock for Stage i + 1 pulses once after c_i pulses of the clock for the previous stage, stage i. This means that each output bit from a stage that is shifted into the next stage's cascade register is a function of a completely different selection of bits in its own cascade register.

Consider an example in which s is 36 and there are three stages. c_1 is 4, c_2 is 3 and c_3 is also 3. The selection clock pulses 36 times faster than the feedback clock that shifts the feedback register. The clock for the second stage pulses 4 times slower than the selection clock which makes it 9 times faster than the feedback clock. And the clock for the third stage pulses 3 times slower than the clock for the second stage which makes it 3 times faster than the feedback clock. Each of the slower clocks can be cheaply derived from the next faster one.

Continuing with the example, each of the 36 selections from the feedback register for a bit of the encrypting sequence is shifted into the first stage's cascade register. After every fourth selection, an output from the first stage is shifted into the second stage's cascade register. We can partition the 36 selections for an encrypting bit into nine sets of four selections each with each input to the second stage depending on one of the nine sets. After every three of those sets, an output from the second stage is shifted into the third stage's cascade register. Another partition of the 36 selections into three sets of twelve each gives the selections on which each input to the third stage depends. When the feedback clock pulses, the output from the third stage which is a bit of the encrypting sequence, depends on three bits in the third stage's cascade register which depend in turn on all 36 selections.

The cascade of stateless functions remains a stateless function itself despite the cascade registers. This is because each cascade register is filled completely at least once between feedback clock

pulses and the inputs that are shifted into a cascade register while producing a bit of the encrypting or decrypting sequence depend only on feedback register bits selected since the last pulse of the feedback clock.

Returning to the example with three stages, it is easy to see what the equivalent stateless function would be in Figure 2. The selection register would have 36 bits. There would be nine copies of the first stage's function, each one taking its inputs from four bits of the selection register. There would be three copies of the second stage's function, each one taking its inputs from three of the nine first stage functions. Finally, there would be one third stage function whose inputs come from the second stage functions. Its output is the encrypting sequence.

CONCLUSION

The cipher feedback systems presented provide a way for some applications that need self-synchronization to suffer less from the tradeoff between the security and error propagation problems common to all cipher feedback systems. Applications in which ciphertext noise varies or lower plaintext noise than the maximum tolerable has value can benefit if they have messages that vary in the degree of cryptographic security needed or if the value of an aggregation of messages exceeds the value of the individual messages.

The effective length of the cipher feedback register can be changed from one message to the next simply by agreeing on a new noise control value which need not be secret. Shorter effective lengths can be used for less valuable messages so that they will be corrupted with less noise.

The use of two different clocks allows the system to have a stateless function as necessary for self-synchronization. With even more clocks, a cascade can be used that helps to keep down the cost of the system.

###