UNIVERSITY OF
COMPUTING
LABORATORY
NEWCASTLE UPON TYNE

THE PROTECTION OF DATA BY CRYPTOGRAPHY

D.W. Davies

This paper is an extract from the NPL Report by D.W. Davies and D.A. Bell entitled "The Protection of Data by Cryptography", January 1978, (COM 98).

Cryptographic Capability

Traditionally the meaning of cryptography is 'hidden writing' and the methods used to hide the content of a written message can also be applied to data. In our present context we mean the transformation of a data message, or a data stream, by means of an algorithm so that anyone observing the transformed data cannot deduce the information which it hides.

Cryptography has a long history but the earlier and traditional methods, with one exception, are too insecure for present needs so we shall not dwell on the history nor indeed discuss in any depth the technique of cryptography itself. We base most of the paper on the publicly announced US 'data encryption standard'.

The Need for a Key

The earliest and simplest cryptographic schemes relied on the secrecy of the algorithm itself. This is not good enough because once the algorithm has been discovered all communication is insecure. Any useful system therefore employs both an algorithm and a key. The key changes the nature of the algorithm so drastically that, in effect, the transformation of data which comes about for each separate value of the key is entirely different in nature. Then if the algorithm is compromised and even one value of the key is discovered, by changing the key the secrecy is restored. We have a 'class' of algorithms as big as the range of keys.

Figure 1 shows schematically a communication channel employing cryptography. The encryption algorithm works on the incoming 'plaintext' and uses the chosen key to produce the 'cipher text' which is transmitted. The aim of cryptography is that access to this transmitted data gives no clue to the plaintext. At the receiving end a decryption algorithm makes the inverse transformation, using the same value of key.

The transformation from plaintext to cipher text must have special properties. Given the correct key it can easily be inverted, but not without the correct key. The cipher text can also be regarded as a function of the key, and this is a "one-way function". For such a function, C = f(K) the value of C is relatively easy to calculate K. The cipher text must be as long as the plaintext, or the cryptogram could not be inverted.

The properties of the cryptographic algorithm can be summarised by the table below in which possibility of deducing (without excessive computation) one of the three quantities, Key, Plain and Cipher from one or two of the others is listed as "yes" or "no".

	Key	Plain	Cipher
1	Known	No	No
2	No	Known	No
3	No	No	Known
4	Known	Known	Yes
5	Known	Yes	Known
6	No	Known	Known

Table 1. Can a quantity be derived from a knowledge of the others? - six cases

If we study this diagram we see that every part of the system, except for the cipher text, must be kept away from prying eyes, or electronic devices. Signals coming from the clear text must not leak onto the cipher text line, even in small amounts, and they must not be radiated, visible with binoculars or accessible in any other way. Suppose, for example, that cryptography is associated with an intelligent terminal, then the terminal software might slip apparently insignificant changes into the headers of messages which, over a period, give clues about the value of the key to someone tapping the transmission line. So every aspect of the apparently secure equipment which generates the cipher text must be examined for possible leaks.

The most important of all is to keep the key secure. The figure shows that the key itself must be transmitted from one station to the other and this is a communication link without cryptographic protection. A key must travel by means that are physically secured. Distribution of the keys and their physical security can be a large part of the cost of implementing cryptography.

Keeping the algorithm secret will make the cryptanalyst's task more difficult but it is the change of key which really gives security. This being so, the strength of algorithms is nowadays assessed on the assumption that the algorithm is known to the enemy. In the case of a publicly announced "standard algorithm" it is a fact that the enemy knows it but this makes little difference to the strength of modern systems.

The Known Plaintext Attack

It might be thought that the assumption that the algorithm was known made life hard enough for the designers of cryptographic algorithms. Modern systems demand an even greater degree of impenetrability for they expect that even a knowledge of the plaintext and its corresponding cipher text does not enable the key to be deduced with a reasonable amount of work. The technical term for using rhis knowledge is "known plaintext attach". Consider how the plaintext might become known.

In a commercial situation a company might put in a long and complex bid with the knowledge that this must be passed, verbatim, to the head office for consideration. The exact length of the message might indicate which of a number of transmissions was the one containing the bid and, by observation of the transmitted signals, we have a known plaintext attack. It was recounted that, during the war, an aircraft might be sent to carry out an unusual attack, such as on a lighthouse, in order to generate, with high probability, a message which could be observed in its encrypted form and for which not too many different plaintext versions had to be tried.

Given a plaintext and the corresponding cipher text it becomes possible, in principle, to determine the key by trying all possible key values to see which one fits. This might also work if we had available, not the plaintext itself, but some strong statistical or combinatorial property of the plaintext. Trying all the keys is the familiar method of attack on the security of car door locks where only a hundred or so need be tried. For sensible cryptographic schemes numbers are very large. There are worked examples in the next chapter.

The Public-Key System

The general scheme of present day cryptography is shown in Figure 1. It employs the same secret key at the sending and receiving stations. A different scheme is shown in Figure 2. Here the keys used at the two ends are different, and both are derived as functions of the common 'starting key'. One special case is that this starting key is itself one of the two keys and the other is a function of it. Following the established principle that the enemy may discover the algorithms, we assume that the key generator functions are known. They may even be published as part of a "standard algorithm".

Suppose that such a scheme can be derived, what is its advantage? If we suppose further that, given one of the keys, it is not feasible to calculate the other then we might make the first key public. In the figure, the key used at the sending end is the public key. This must be a "one-way function" of the starting key and it must be impossible to deduce from it either the starting key or the other, secret key. Finding the secret key by trying all possible starting key values is ruled out by the sizes of keys.

Such a 'public-key system' has the great advantage that secret transmission of the key is no longer needed. Someone who wishes to receive secret messages can announce his public key while keeping the corresponding secret key to himself. Anyone else can then encrypt a message for him and only the authorised receiver can decrypt it. Note that the public nature of the key loses us the property of "sender authentication". If no other precautions are taken, one sender can masquerade as another. The secret key of the 'classical' system, if kept secure to its intended users, does help to authenticate their messages. We shall see later that a second public key can be used for this purpose.

The Des Block Cipher Algorithm

The invitation to develop a suitable algorithm was first issued in March 1973. IBM put forward a system which was published in 1975 and adopted by the Institute for Computer Science and Technology in the NBS as a proposed Federal Information Processing Standard. It has been referred to in the literature variously as "the NBS algorithm", "the IBM algorithm" and "the Data Encryption Standard" or DES. It was published in its final form on 15th January 1977 and became effective as a standard 6 months later. A Federal Information Processing Standard (FIPS) is mandatory for U.S. Government civil purposes except where there are special reasons to depart from it.

The Data Encryption Standard (DES) is a complicated series of permutations and substitutions which are applied repeatedly to the message. It accepts a 64-bit input block and a 64-bit key (8 of them are parity bits) and yields a 64-bit block of ciphertext. A very nice feature of the DES is that the same algorithm, with one small change, will also serve to decrypt the message.

Application of Cryptography for Data Transmission

The U.S. Data Encryption Standard treats each block of 64 bits separately. If the text can be divided naturally into 64 bit blocks the scheme shown in Figure 3 produces the cipher text and then transforms it back into plaintext at the destination. For a full-duplex channel there must be an identical set of hardware for use in the opposite direction. Probably a different key will then be used for the two directions. This is slightly more secure and not much more difficult to organise.

The independence of each block in this block encryption method has its advantages but also a significant weakness. Suppose that longer messages are split into 64 bit blocks for encryption then, without knowing the contents of the message, an intruder could delete blocks, insert blocks from older transmissions or change the order of blocks before they are delivered to the destination. It might be possible to make a meaningful message by these tricks. For example, the message might have a well-known format and one block might contain some critical information, such as the cash value of a money transaction. This is the block that would be changed, using a block from the same position in an earlier message.

Consider also the transmission of the incomplete block which remains after the full 64 bit blocks have been sent. If this is a small piece it would be unwise to pack it out with zeros or any other fixed pattern before encryption. Remember that the block encryption is still no more than a substitution cipher. What makes it strong is the large 'message space' of 2^{64}, but if most of the pattern is fixed, the remaining small field presents a substitution cipher of its own which would allow a cryptanalyst to start work on identifying what each of the comparatively small number of possibilities means. This weakness does not only apply to the short piece at the end of the message but also to the kinds of format which computer systems use in which certain fields are little used or tend to have constant values.

To avoid these weaknesses some method must be found to introduce interdependence between the blocks which make up a single message.

Chaining of blocks

Figure 4 shows a scheme which has been proposed. We show the contents of registers by means of rectangles but it will become clear that these are not physically different registers; they are re-used with different contents. To simplify the picture, block encryption and decryption are shown by arrows with C or D written across them.

In the first block a small field (shown as a) is reserved for an authenticator which relates to previous messages and identifies this as correctly one of the message sequence. In the cipher text the same area of the block is called 'b' because of its later significamce. The whole of this block of 64 bits is sent to the receiving end where it is decrypted and the authenticator is checked. At the same time the field containing b in the cipher text is retained for future use.

At the sending end this field b in the cipher text becomes the data which chains the first block to the second and this is done by placing it in the corresponding place in the new plaintext. (Ignore the plus signs which have a significance to be described later.) Note that the contents of this same field in the new cipher text is repeatedly transferred to the plaintext of the next block until the transmission is complete.

At the receiving end, the decryption of the second block produces the value now shown as b+ and the receiving system can carry out a check since it has stored the supposedly identical value sent with the cipher text of the first block. This is shown by the line leading from b to b+ on the right-hand side of the diagram. In the same way the value c from the cipher text of the second block can be compared with c+ from the plaintext of the third block. Thus the chaining procedure carried out at the sending end gives rise to a corresponding checking procedure at the receiving end.

Since the effect of each encipherment process depends on the whole of the block, the values of the fields described will be unpredictable and will ensure that no special properties of the incoming data reveal themselves in the cipher text. The problems of block manipulation will effectively be overcome. It is merely necessary to ensure that the first block with the authenticator takes on sufficiently variable values. If necessary, part of the field of this first block could be occupied by a random or pseudo random number.

When the transmission is full-duplex the return path must deal with chaining in itw own way, but for the half-duplex case it becomes possible to continue the same scheme of chaining in the reverse direction, as the diagram shows. The field d from the cipher text of the last block sent in the forward direction is substituted in the same field of the plain text which is due to return in the opposite direction. The figure shows how d and d+ can be compared on the left hand side.

An advantage of this scheme is that any attempt to interfere with transmission will show up by a failure of the chaining field to check at the receiving end. It is inherent in this scheme that any transmission errors will also cause the checks to fail and the whole message is lost. In a packet or message switched network the error control procedures on the link should make this a sufficiently rare event. The procedure occupies part of the space in the block. If 8-bit chaining is not considered secure enough it may be necessary to give up two octets or 25% of the capacity.

A second chaining method can be devised in which this 25% of capacity is not lost. This is the reason for the plus signs in the figure. If we now imagine that the field b is not substituted for the corresponding field in the plaintext but is added (modulo 2) to this part of the plaintext it can be seen that the whole or the 64 bits of plaintext is recoverable at the receiving end. Where the checking arrow passes from b to b+ we now add the contents of the first register (modulo 2) to the plaintext of the second block and this will recover the whole 64 bits. If the figure is examined it will be seen that this works in each case.

In return for the extra usable space in the block we lose the possibility of checking at the receiving end, but in computer transmission there is ususally some other error checking mechanism in operation. The effect of an error in a single transmission is shown by the triple dots in the figure. An error of transmission affecting the block containing c propagates into the received plaintext for the second and third blocks.

Serial numbers in place of chaining

We introduced chaining to overcome the problem of whole blocks in which the variability was slight and which might therefore be susceptible to analysis. But if we believe that the encryption algorithm produces patterns which are a function of all the bits of the plaintext there is no reason to bother with a chaining method at all. The field which was used in the first chaining method could simply be used to carry a serial number, a number which it incremented from one block to the next. Each block must be different from all its predecessors and this difference in the cipher text is as complete as it could be, even though it is achieved only by a serial number.

The property of the serial number makes it about as good as known plaintext to the cryptanalyst. We simply try the candidate key against successive blocks and see if the serial number increments. The work is, of course, doubled. We should not allow the serial number to overspill and start again from zero because this would let the cryptanalyst operate on several blocks of the same serial number and start his statistical work. Therefore, to be very careful we should change the key before the serial number spills over. We also need a method of recovering the correct sequence if there are transmission errors or deliberate interference with transmission, but this is not difficult to devise.

The first chaining method is just about equivalent in performance to the use of a serial number field. It seems that the additive chaining method is preferable since it uses the whole of the block for data.

Stream encryption

With the precautions we have described, block encryption is suitable for store-and-forward communication systems or other block-oriented schemes. Polling systems and data link control procedures in general, including HDLC fall into this category.

There are other data communication requirements in which individual small units such as bits or octets must be transported as soon as they are ready. Generally speaking, communication with simple terminals is like this since they handle individual characters in 7 or 8 bit units. For these non block-oriented transmission shcemes we need a new way to use the DES block encryption algorithm.

Clearly it would not be satisfactory to put one character into a block and fill out with a fixed pattern. This would be a subsitution cipher. Filling out with random bits is one possibility.

The classical method of stream encryption is to add (modulo 2) a pseudo-random sequence to the plaintext at the sending end and then add the same sequence generated by a similar mechanism at the receiving end. The generation of the pseudo-random sequence needs some care and the typical 'feed-back shift register' devices are not acceptable because of their linearity. The DES itself can be used to generate a random stream by feeding back the output to the input and this should be a suitably random sequence.

Cipher feedback

This alternative method employs the configuration shown in Figure 5. At both the sending and the receiving end the block encryption device is used in the enciphering mode. At the sending end a feedback loop is used, corresponding to a feedforward scheme at the receiving end. Because of the feedback, even with a rather fixed pattern of input the cipher text will tend to be random.

Transferring attention to the receiving end it can be seen that precisely the same sequence of octets enters the modulo 2 adder and therefore the plaintext is restored on the output.

It is interesting to compare this scheme with the chaining of blocks in Figure 4. The feedback loop exists in both but the cryptographic device is a different part of the loop.

Now consider the effect of an error on the line. Supposing that it affects just 1 byte then the erroneous byte will move up the 64-bit shift register at the receiving end and eventually spill off the top. While it remains in this shift register the byte emerging from the encryption device will be completely in error. Therefore, the effect of a single error on the line is spread over the size of one encryption block (64 bits) plus one transmitted byte. After this time the system goes back into correct operation. This is sometimes called 'self - synchronising'.

A similar process occurs when the system is first started up. The contents of the shift registers at the two ends of the line are initially different but they resynchronise after the appropriate interval.

One of the special problems of computer data is that the first messages of a transmission may take a constant form. Therefore, it is essential to start up the cipher feedback device in a different way each time it is reinitialised, certainly if the same key is used as was used on the previous start-up. This is best achieved by loading the shift register with a random or pseudo random sequence of bits.

Suppose that we wish to make an instant start and begin transmitting data from the first byte, then a pseudo random sequence must be used which is known to both ends. To preserve synchronism the values of the algorithm generating this sequence will have to be held even when the equipment is switched off.

It will be seen that a known plaintext attack is little affected by cipher feedback. From the DES point of view all the 'plaintext' input is known as well as a part of the 'cipher text' output. 64 bits of text is sufficient material for this attack.

Authenication of messages

To authenticate a message we need it to contain, in a concealed form, a personal identification number (PIN) which belongs to the sending authority. At the same time we need to ensure that the message cannot be changed nor the PIN discovered in transit. This is analogous to a bank cheque in which the contents of the message are free for everyone to see yet supposedly it cannot be altered in transit and the signature shows that it was authorised by a certain individual. Figure 6 shows how this can be done.

The encryption device operates on all the context of the message which must be protected; if a message is a long one this may involve repeated application of the block encryption algorithm. From the result, a 'residue' is formed of sufficient length to form eventually the 'message authentication code' or MAC. (This is otherwise called a cryptographic check sequence.) Its size must be large enough not to be obtained by luck and 16 bits will probably be the minimum.

Sixteen bits from each of the operations of the block encryption device are sufficient and they can be added together to form the residue. It is sufficient that each bit of a residue is a function of the whole of the message which it authenticates.

To form the message authentication code we add the residue (modulo 2) to the personal identification number. The MAC is then appended to the message for transmission. Now anyone can read the message and authentication code but someone who knows the key employed and the PIN can check whether the MAC is valid for that particular message.

This form of message authentication by cryptography has one snag. The receiver of the message must know the key and the PIN if he is to check the authenticity, yet this would allow him to forge a message. Because of this, in the very untrusting environment we are assuming, the sender could falsely accuse the receiver of forging a message that was, in fact, sent. Instructions to make a speculative transaction that subsequently failed could be denied. A method is needed by which there is public proof of the origin of a message.

Message authentication using the public key system

You will recall that the public key system of cryptography is a theoretical possibility and there has been a recent proposal that seems likely to yield a practical system. Supposing that a secure and convenient public key system has been established then, with certain conditions, it can provide 'unforgeable' authentication.

Figure 7 shows the scheme. Comparing it with the cryptography scheme we note that "decryption" and "encryption" are in their wrong places. This is possible if the transformation employed did not increase the number of bits. In that case, the n possible values of an n-bit plaintext block receive a one-one mapping onto the 2^n values of the cipher text. The decryption is the inverse of this mapping. Clearly the sequence of the two inverse one-one mappings can equally well be "decryption" followed by "encryption". An alternative possibility is to leave decryption and encryption in their usual places but this changes over the function of secret and public keys. Perhaps the system will allow this while still preventing calculation of the secret key as a function of the public one. But this alternative implies that a different pair of keys is used for cryptography and authentication. The scheme in Figure 7 is better because, having calculated and published his secret key a user can authenticate his own messages with it and receive messages from others encrypted with it. So a public key system should if possible, have the one-one mapping property.

The transformed text is not secure because anyone, using the public key, can "encrypt" it back to plaintext. But it does contain unforgeable evidence in its origin because only the sender who has the corresponding secret key could have produced it. The public key and the transformed message can be produced as evidence that the sender (if he was not careless with his key) actually sent it. A court of law would also be interested in the contents of the message, which are a demonstrable consequence of the key and the transformed message, and hence are linked incontrovertibly to the public key.

As a further elaboration it might be necessary to authenticate a secret message and this can easily be done by using the public and secret keys of sender and receiver, as Figure 8 shows. Here the two users of the public key system (cryptography and authentication) have been tested. Suppose now that the keys are still in use and the receiver wants to prove that he received a certain authentic message. He need only reveal the form of the message at X and the sender's public key and show that this combination produces the plaintext. It is unlikely that the identity of the sender would be of interest without being linked to the message content.

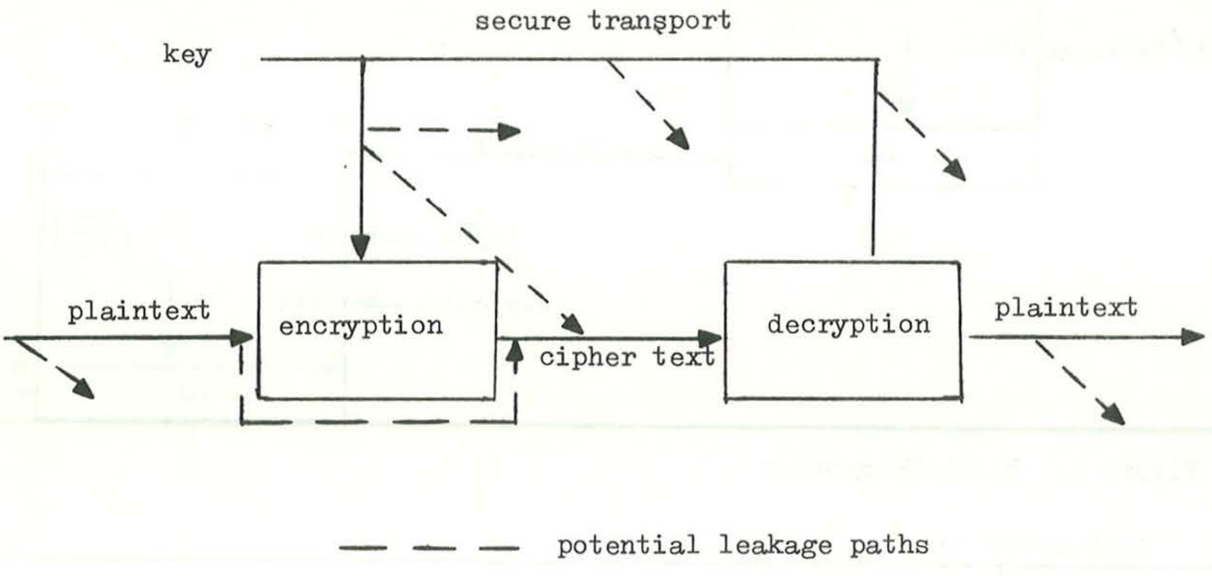

Figure 1: A Cryptographic Scheme

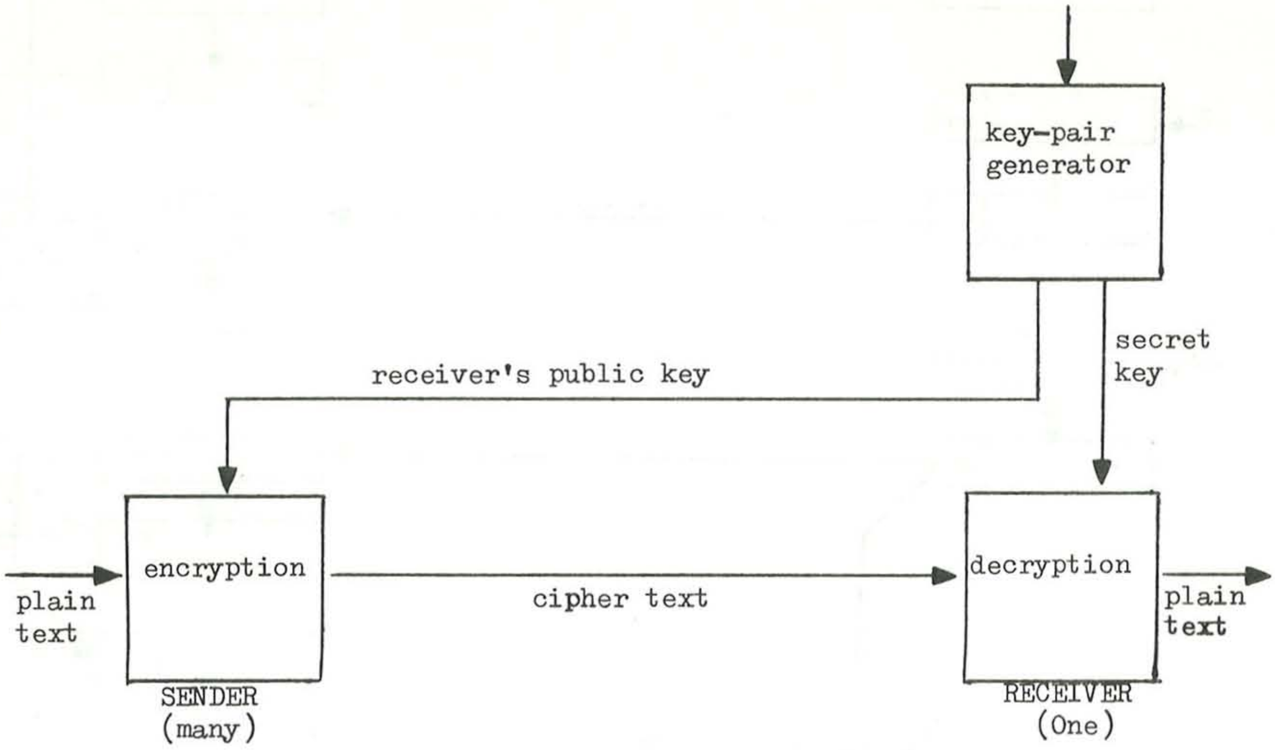

Figure 2: The Public-Key Cryptosystem

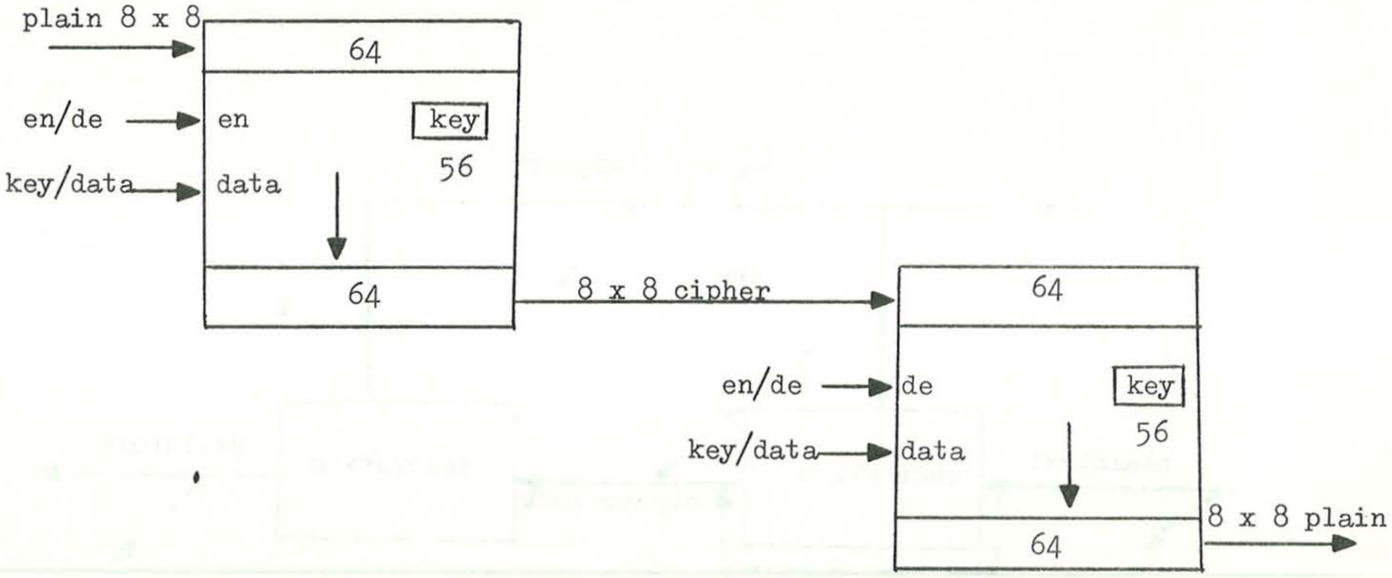

Figure 3: Block Encryption

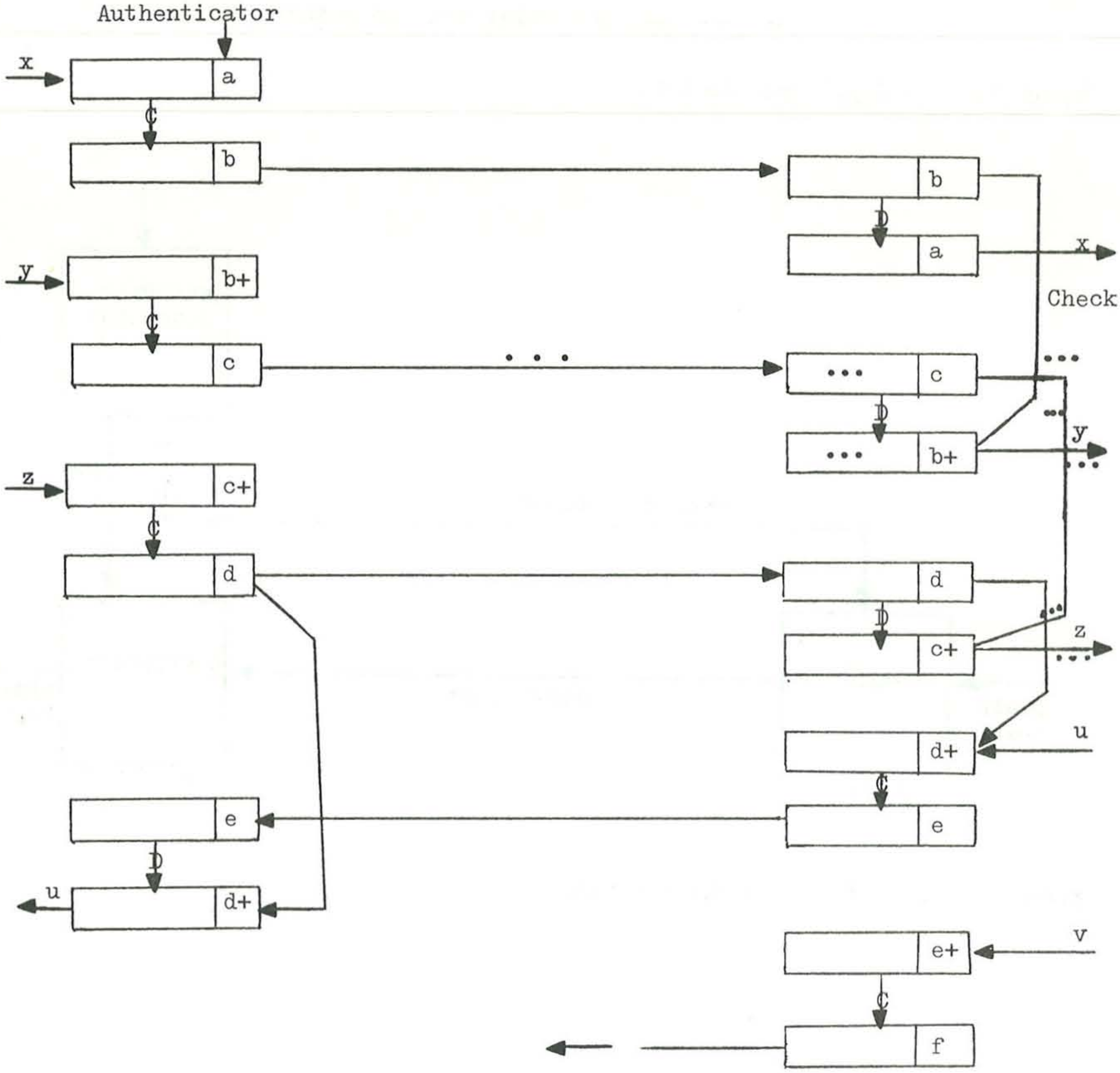

Figure 4: Chaining of Blocks - Half Duplex Example

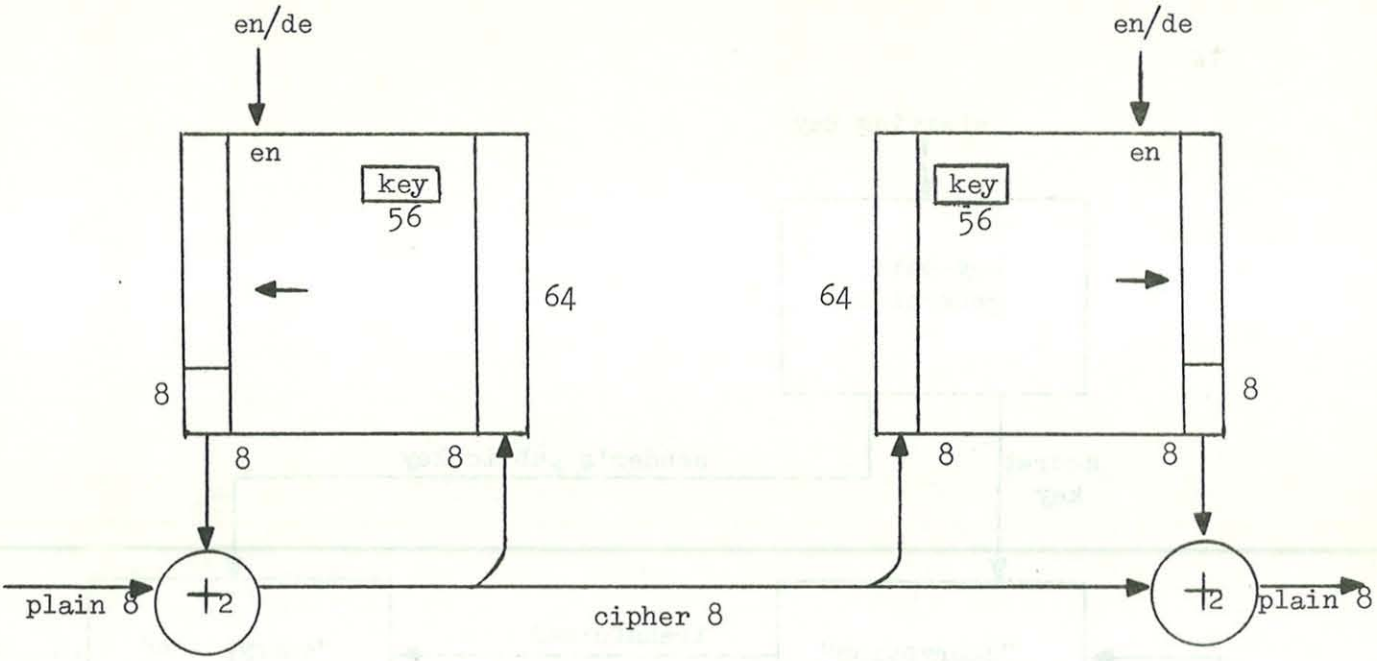

Figure 5: Cipher Feedback

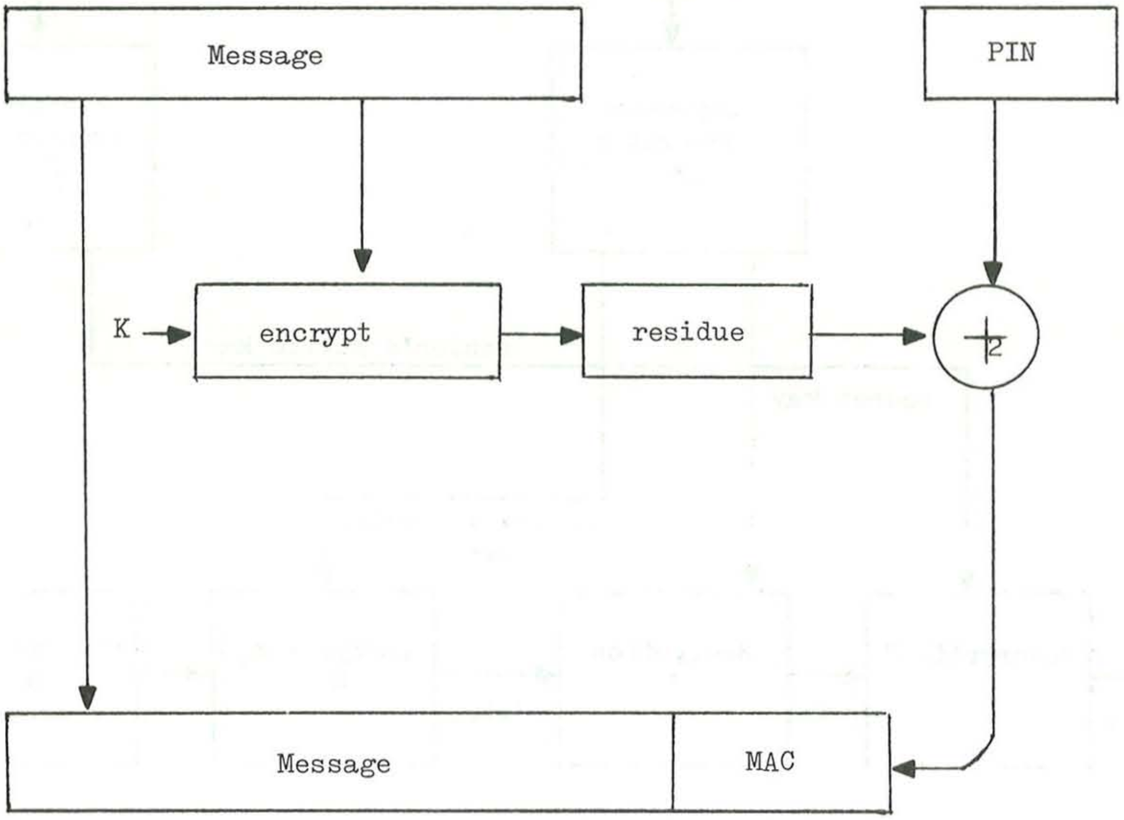

Figure 6: Message Authentication

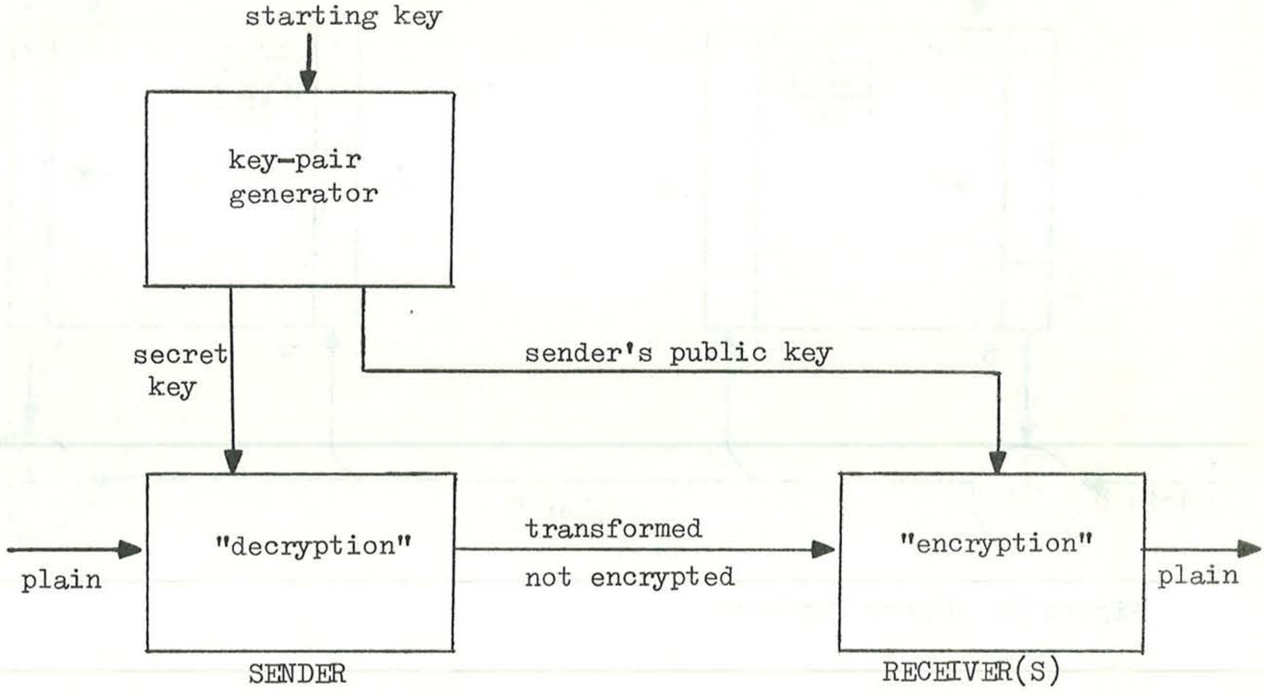

Figure 7: Message Authentication by Public-Key System

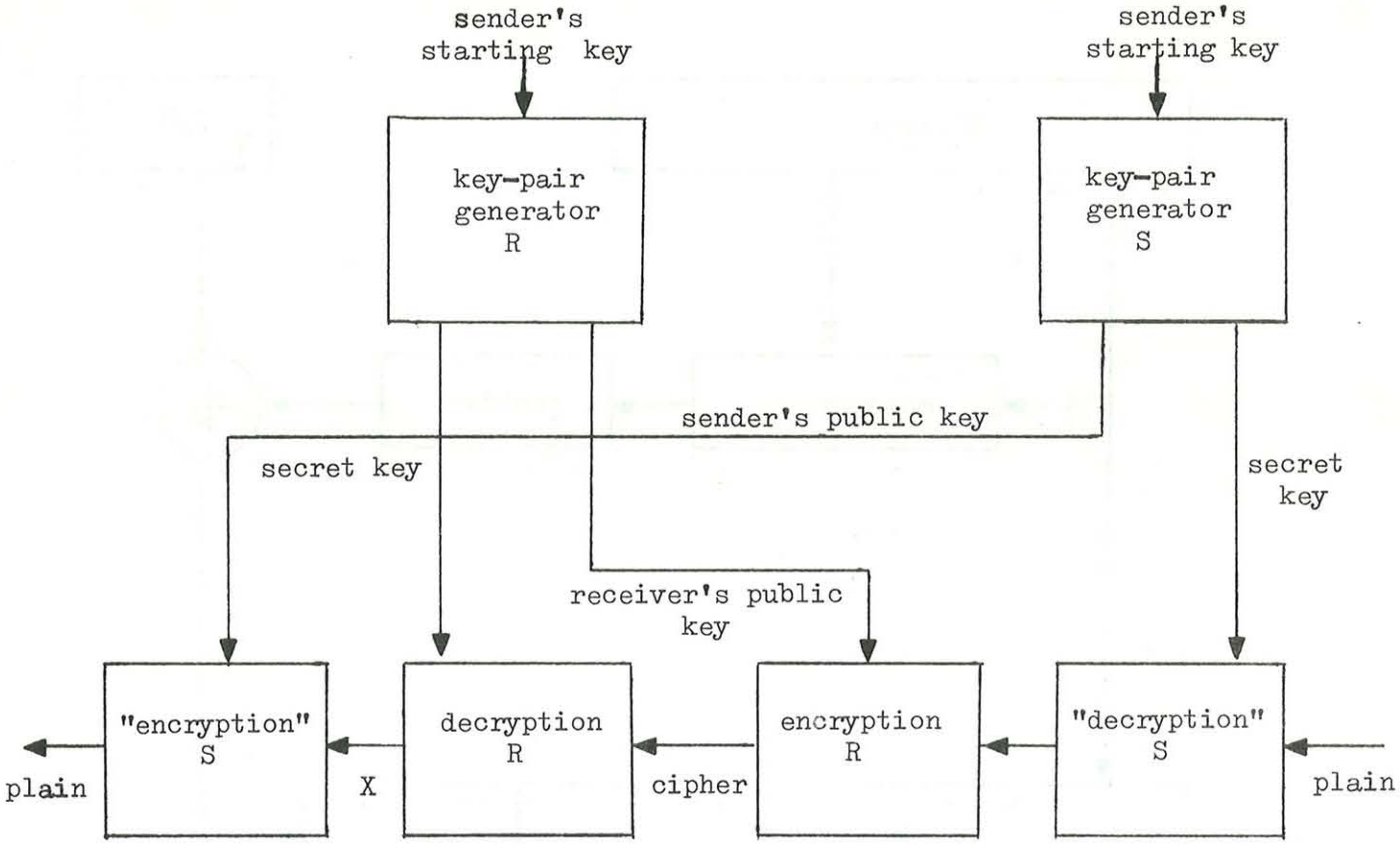

Figure 8: Combined Public-Key Cryptography and Authentication