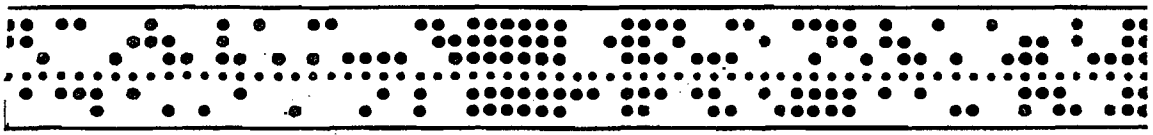
FOURTH LECTURE: One-Time Tape Systems

So far in these lectures, all of the systems I've mentioned have had one thing in common. They have widely differed in structure, process, security and application: the thing that has been the same about them is their relation to the communications process. They are all *off line* which means, once again, that they work essentially independently of the communications set-up; they are not tied into the communications path; the complete encryption process is performed *before* the cipher text is transmitted, and the nature of the communications system to be selected for the eventual transmission is not of much consequence.

From now on, with a few exceptions, the systems we will be talking about will be more and more involved with specific means of transmission; most of them will be on-line systems or systems with both an on-line and an off-line capability. This means that the machines themselves, or the ancillary equipments used with them will be more and more tailored to particular communications techniques and eventually, as you'll see, will involve the *integration* of the cryptographic process into the communication system itself.

The first and simplest set of systems lashed into their associated transmission means are the one-time tape systems. They are called the PYTHON systems for fairly obvious reasons. From World War II until about 1960, these systems were very popular indeed, and are still rather widely used. In both WW II and the Korean War they formed the backbone of secure U.S. teletypewriter communications. I can name more than 12 different machines built since 1945 for PYTHON operations. Their principle is deceptively simple, you merely take a stream of random key in *binary form* and *add* it—*combine* it, *mix* it—element by element, with plain text that has also been produced in *binary* form. To put intelligence into binary form is to convert it (or, in the generic sense, *code* it) into symbols made up of only *two* elements—1's and 0's—the familiar computer language; or pluses and minuses, or on's and off's, or marks and spaces or, as on tape, holes or no holes as indicated in the following illustration:



Various teletypewriter equipments automatically convert characters into this binary form, for example, in the Baudot teletypewriter code:

A = + + − − −; R = − + − − −, etc.

The additive or mixing process is done according to a simple, arbitrary rule: like signs = plus; unlike signs = minus. Now, let's add:

PLAIN TEXT	− + − − −	− + − + −
RANDOM KEY	− + + − −	+ + − − −
RESULT (CIPHER TEXT!)	− + − − +	− + + − +

It turns out, that if you take the same key and add it in the same way to the cipher text, the resultant product is the plain text again—and thus you decipher. If you can find a way to do this mixing mechanically, or electrically or electronically, you can visualize an extremely simple set-up. Your send and receive machines are identical and use identical key tapes in identical ways.

You do not have to reverse your process, switching everything so it goes backwards as we did in the rotor machine. The receiver merely assures that he is using the same tape as the sender, and has started it in the same place, and by adding it to the cipher text he has received, gets a copy of the original plain text printed automatically for him by the teletypewriter equipment.

Like all other one-time systems, though, the key must be used once and only once for encryption; if it's good random key and is used properly, the cryptographic security seems to be absolute. If you use the same key twice for encryption, the security drops to approximately 0, forthwith.

I said I could name about a dozen of the machines. The reason for the variety stems from two causes: first, the adaptation of machines to more and more refined concepts of teletypewriter communication; second, the need to prevent compromising radiation—the electronic emission of intelligence in the form of radio frequency energy from the various switches and contacts and relays in the equipment. We'll talk about *that* problem at some length in the last lectures.

The simplest kind of teletypewriter transmission path is a line from point A to point B with transmissions travelling in one direction only. This is called a *simplex circuit*. There are some obvious disadvantages: B can't talk back. A much more common type of circuit is a path between A and B on which either station can send when the other is silent. This is called a *half-duplex* circuit. Still some disadvantages: they both can't send at once—something communicators like to do, especially if each has a high volume of traffic for the other. The optimum setup permits transmission to flow in both directions simultaneously and is called a *full-duplex* circuit. Such circuits really involve two separate radio paths or two pairs of wire lines, but some of the terminal equipment may be shared. Different kinds of one-time tape crypto-equipment were envolved to fit with these differing communications setups.

The simplest way to send teletypewriter characters over the paths is by what is called "Start-stop" operation. The receiving machine *waits* until it receives a character, deciphers it, moves its one-time tape one position, and waits for another character before operating again. So it keeps *in step* with the sending machine by using each actual cipher character received as a signal to advance. Most of the old one-time tape mixers worked this way. But suppose the transmission fades momentarily, and the receiving machine misses just one character; or suppose some spurious pulse hits the signal line and causes the receive machine to advance when no cipher character was really sent? Then the two machines are out of step—synchrony between send and receive tapes is lost, the keys no longer match, and thereafter the receiver deciphers gibberish until the operator can signal the other station to stop and they get themselves in step again. So they began to design machines which would step along at a fixed rate once they got started together, whether every character was received or not, and the short transmission fades or spurious pulses simply caused a one-letter garble in the received text. These are called *synchronous* machines, and account for two or three more of the dozen mixers that have been in our inventory.

Yet another feature became desirable for some one-time tape circuits. You will recall that I have mentioned the term Transmission Security or "TRANSEC" just once so far. We were discussing a manual one-time system and I alleged some COMSEC shortcomings despite its great resistance to cryptanalysis. The bread and butter of *transmission security* specialists is the information that they can glean merely from analyzing message *externals* as they are transmitted. Call signs tell them something, so do routing indicators, so do cryptographic indicators, so do the numbers and lengths and formats of messages, so does the direction in which traffic flows. If the government is planning a secret operation in some remote or not so remote place, there is almost bound to be a great spurt of message activity to and from that place, and all the opposition need do is note this surge of communications activity to be put on guard. The technique which we now commonly use on teletypewriter links to remove most of these flags on impending activity is called *traffic flow security*. In a one-time tape setup, the way it is accomplished is to simply send cipher text or something that looks exactly like cipher text *all the time*. Instead of cipher characters being transmitted by fits and starts only when an operator is actually typing a real message, or where a few hundred groups are coming out in a stream if the operator is sending his message automatically from a previously punched message tape, the machine is rigged so that whenever an actual mes-

sage is not being sent, the successive characters of random data on the key tape itself are automatically sent instead. So the roll of tape just sits there and unwinds all day, encrypting anything you happen to have for it and being transmitted itself otherwise. The tape on the other end is doing the same thing, of course. All the interceptor sees is an apparently continuous flow of random information. What does the receiver see? Since his tape machine tries to decrypt anything it receives, it winds up decrypting *key* when no bona fide traffic is coming in. Let's have a look at what any one-time key decrypted (i.e., added to itself) looks like. Remember our rule—like signs = plus; unlike's = minus.

```
++--++++++-++++--
++--++++++-++++--
----------------

++++++++++++++++++.... All pluses!
```

And all pluses equate to the letters shift character in the Baudot code, and it's a relatively simple matter to instruct the teletypewriter to stop operating until it gets something else. Otherwise, you can just let it run. So, equipments with this traffic flow security feature account for a couple of more of our many PYTHON machines.

Well, let's have a look at the advantages and disadvantages of these PYTHON systems. The first advantage is relatively great speed compared to any of the systems we have described so far. In most of the manual systems you feel like a whiz if you can average four or five words a minute: in our off-line rotor machines, we were happy with 25 words a minute and simply couldn't go much more than 40. But a PYTHON system operates at standard teletypewriter speeds—66 or 75 or 100 words a minute. And besides, when you're *on-line*, the message is being received instantaneously at the distant end: so with PYTHON we are moving toward the goal of secure communications in which no delay in message delivery can be attributed to the cipher process itself. You're still consuming a little time in pure cryptographic processes—you have to select and set up the proper tape; you have to send an indicator of "Set" to the distant station to tell him what tape to use and where to start it; but most of the time is spent in preparing the message for transmission—punching it up on a message tape ("poking" they call it) before feeding it into the machine—this is something you have to do anyhow for efficient teletypewriter communications in any volume. So, on the matter of speed, we have made a great leap forward.

The second advantage is its relative simplicity: most of the system consists of standard time-tested teletypewriter machine components which are commercially available; maintenance is relatively easy; teaching an operator to work the system is simple; mistakes are hard to make and only one mistake—the reuse of a tape—is dangerous to the security of the system. (In contrast, on a system like KL-7, there are a dozen or more things that operators can and do do wrong which give us grey hairs.) There are other things that can go wrong of course; technical things, like the tape getting torn and failing to feed properly and the machine going merrily on encrypting all of the message using whatever key character the tape happened to stick at—monoalphabetic substitution again! But there are a number of safeguards built in for contingencies like these, and by and large it is safe to say that a typical one-time tape system is both reliable and highly secure.

So, the advantages, in summary are: fast, simple, reliable, and secure. How about the disadvantages? By now, the first disadvantage ought to leap readily to mind. They are one-time systems, and the inherent disadvantage in all of them applies here. Only two or a few more holders can intercommunicate in a given system—we make a few "five way" tapes and "ten-way" tapes to accommodate some broadcast or conference type teletypewriter communications; but it's a difficult job to get everybody in step and keep them there, and by and large the two-holder or "point-to-point" system prevails.

The second disadvantage is a logistic one: imagine the complexity of the distribution system that gets thousands of pairs of these tapes out, to holders all over the world. Their bulk, in a large communication center in which many tape systems terminate, is staggering. In their heyday

[redacted]60,000 rolls of tape were produced by us in 1955. Production is around 55,000 now. The con[redacted]ption of these tapes is particularly distressing when that transmission security feature—traffic flow security—is employed. One of these eight-inch 100,000 character rolls lasts about 166 minutes at 100 words per minute; they cost us $4.55 each.

At any rate, their usage has begun to decline sharply as more efficient means for doing the same job have evolved. As early as 1942, the people designing cryptomachines had tried to come to grips with the logistic problem associated with one-time tapes. All the one-time tapes used by the U.S. come right out of Operations Building #3 in what is called a tape-factory. Great batteries of tape generation equipment, which will be described to you later in the lectures on the production process, can spew these tapes out at the rate of thousands of three-inch rolls per day. In the old days, the manufacture of these tapes was slower. Very large machines were used to produce carefully checked random data to be punched into the tapes. "Suppose," said the cryptographers, "*you could build a machine that could generate its own key as it went along* and feed that key to a mixing or combining circuit electrically without having to punch it up in a painstaking mechanical fashion on a stretch of tape? Give the man at each end of the circuit a *key generating machine* which, from given starting setups, would produce identical key that could be used in this same old binary additive mixing process that works so well with the one-time tape systems. Then, instead of having to distribute carloads of tapes to these people, we would merely need send them a little printed key list containing the settings that should be used for the variables contained in the little key-generating machines."

And that's what they did. They called the equipment SIGTOT in accordance with some old Army Signal Corps nomenclature scheme. It used rotors, and it worked pretty well. Its key output fed into a standard one-time tape mixing machine and got combined there in the regular old way. But it used rotors with all their mechanical difficulties, and we found ourselves shipping around truckloads of rotors instead of carloads of tape. When you see the tape factory, you'll note that a rather massive batch of machinery with all sorts of checks and alarms are used to assure a completely random [redacted]oduct. When you try to compress essentially the same operation into equipment about as big as a [redacted]readbox, you might expect troubles, and we had them. We wound up with all sorts of precedural [redacted]traints on the use of these systems for security reasons, and eventually had to use a set of no less than 30 rotors to support each machine so as to provide an adequate bank of variables to choose from. Still, the SIGTOT, with various modifications, lumbered on in some quantity from WW II until the mid-fifties and the last ones did not disappear until about 1960.

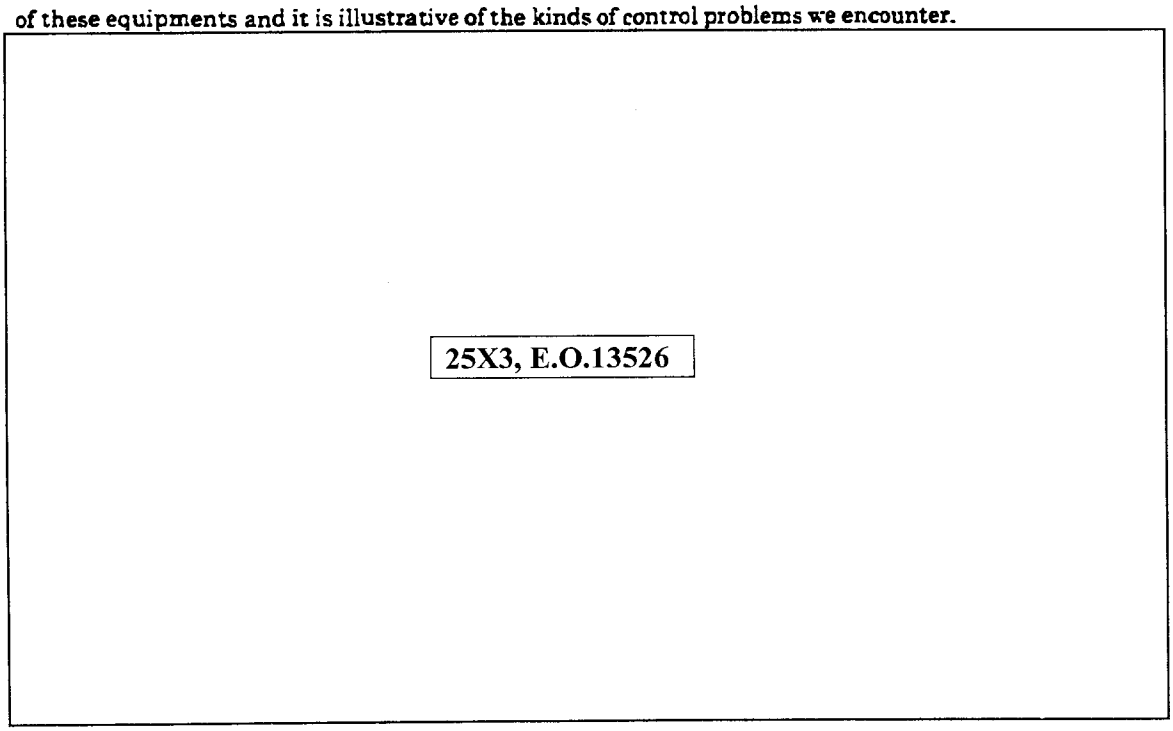
****

So far, we've confined ourselves pretty much to how these various systems work, what they can do, and what they are for. Before we jump into the electronic age of cryptography, perhaps it would be well to discuss some of the things that go into the production and support of a cryptosystem beyond the provision of sound cryptoprinciples and some techniques for making them work—by embodying them in pads or charts or tables or in some kind of cipher machine. Implicit in what I've said already, you have to have somebody design and develop these systems and, in the case of hardware, that's what NSA's R&D COMSEC organization is for. You have to have somebody *evaluate* these designs; and it seems sound practice to have a body of people who are separate, objective, disinterested, do this job—not the inventors themselves who are apt to have prejudices and blind spots with respect to their own brainchildren; and that's what our COMSEC analysts are for. You have to have somebody who can take these approved designs and prototype equipments and engineer them into fully tested working systems that can be produced efficiently and in quantity—to make a finished product which, in addition to being theoretically secure will be economical, reliable, and practical to produce and maintain. That's what the COMSEC Office of Communications Security Engineering (S2) is for. There are still more things you need. You have to have an organization to produce and distribute these volumes of variables on which every one of these systems in one way or another depends. That's what the Office of Communications Security Production and Control (S3) is for, and, of course, you need instructions. You need the specific [redacted]erating instructions that tell operators just what to do, what processes to follow, how to react if [redacted]nething goes wrong; you need systems planners to anticipate and meet requirements and to get

the right equipment applied to the right job. You need a very involved and interlocking set of security controls over the materials and equipments in the inventory—you need to decide how to mark, classify, ship, store, account for, and eventually destroy every item. You need a whole system of surveillance to watch over systems as actually used to assure that they meet their security objectives and, where they don't because something has been lost or some other catastrophe occurs, to implement, and implement at once, whatever countermeasures—like the emergency supersession I talked about—that can be put into effect. This means a world-wide reporting system to inform us electrically of events that may effect our COMSEC posture, and a large quantity of back-up or reserve materials for use in an emergency. During FY-72, the Office of Communications Security Applications (S4) was established to better support the systems approach to COMSEC. This organization consolidates and emphasizes the S effort towards the system approach, wherein security is functionally and physically intergrated into communications-electronics systems of all types. It insures a consistent and coordinated effort in meeting NSA's responsibilities to system designers, developers and users for providing COMSEC support and provides a focal point within S for outside organizations to turn to in seeking assistance in systems matters. And finally one of the most difficult jobs of all—you need a large, consistent, coherent, practical, responsive, safe, reasonable, and understandable body of *doctrine* to govern the whole shooting match, and this is what the Office of Communications Security Standards and Evaluations (S1) and the Technical and Planning Staffs are for. And these are all more or less central functions here in NSA; large counterpart organizations, especially in day-to-day monitoring and administration of systems, are required among the users. For what we are talking about here is the management of a very large operation—not only are millions of copies of paper materials involved, but we are supporting on the order of 100,000 relatively delicate, undoubtedly contrary, tricky, recalcitrant, *classified* cipher machines.

Perhaps you did not realize it, but what I've just done is sneaked in on you a rundown of the functional organization of the COMSEC part of this Agency.

I have implied that the business of protection and control of cryptomaterials constitutes a large and difficult area of endeavor for us. While one-time tape machines are fresh in your mind. I want to discuss classification for a moment, because there is a small controversy about the classification of these equipments and it is illustrative of the kinds of control problems we encounter.



The second reason is clearly a COMSEC one. Even our newest one-time tape mixer is not perfectly secure. I keep titillating you with this business of compromising emanations; we want to keep other people from discovering the techniques we use to suppress these emanations; and we also want to make it difficult for them to find out where we have still been unsuccessful. It turns out that the ideal way to exploit the radio frequency or acoustic emissions from a cipher equipment is to get the thing in a laboratory and test it very thoroughly and minutely to find out in what part of the spectrum, if any—the emissions are escaping and just what their characteristics are. Having done this, you know how to zero in your intercept equipment in the much more difficult environment where machines are actually operating, and your chance of success is much greater than if you have to go at it blind.

There is another related and long-standing notion about classification of crypto-equipment that is worth discussion here. It involves a rather difficult concept, more often misunderstood than not, and one that often causes much anguish among our customers each time it leaks out in distorted or incomplete form. Here it is: whether we're talking about a one-time tape machine, or the KL-7, or a modern key generator system, the essential security lies in the variables supplied with the equipment, *not* in the configuration of the equipment itself—not in its wiring, motion, activity, or processes. This means that if the machine is lost, no past or future messages encrypted by it will be jeopardized unless its variables—its *keys* are lost as well. There's a very practical reason for designing systems this way: no matter how highly we classify an equipment or how carefully we guard it, we cannot *guarantee* that it will not be lost. All of them are designed to be useful for 15 to 20 years and a lot of things can happen in that time—military units can get overrun; planes can crash in hostile territory; people can defect. We simply can't afford to replace 10,000 key generators or 25,000 KL-7's should that happen.

So, in a nutshell, if you lose the equipment, but not the keying material, your traffic is still cure. When the customer hears this, he has a natural question: why in the world do we insist on lassifying these machines then? And he has more than an academic interest: the protection of these machines costs him money and time and guards and vaults and specially constructed crypto-centers and a host of attendant headaches.

Well, why *do* we insist that this expense to the user—and it's a real expense—is a worthwhile security investment? I have already touched on the matter of general exposure of our technology. But there are even more cogent reasons for trying to protect principles and details of machine operation *when we can.* The first is this: although we strive for reliability, and sometimes can afford to incorporate rather elaborate alarms, machines do sometimes fail or partially fail. In the case of a modern high-speed key generator, thousands or millions of bits of faulty key or cipher text may be put on the air before the problem is detected and the machine halted. There may be even more insidious failures that do not affect communicators' ability to encipher and decipher messages, but seriously weaken the resistance of the system to analysis. The discovery of exploitability of such situations by hostile interceptors may well depend on whether he understands the fundamental structure of the machine in use; so denying him that information to the extent we can is important. Similarly, operators may make mistakes that may be harmless if the interceptor does not understand the system, and exploitable otherwise. Note, the basic proposition is still that the traffic is secure with the machine known, but with the keys safe. We have to modify that statement to indicate that this is so except in cases where the machine is operating improperly—and sometimes they do operate improperly. And we have said, there's not much problem so long as the keys are safe. The trouble is we *do* lose keys (in FY-72 there were 325 incidents of loss and unauthorized viewing). But a stolen key will generally not do the hostile analyst much good unless he knows how the machine works that uses it. Finally, the most important reason for protecting machines is that a hostile cryptanalyst generally cannot even make a start on the analysis of any cryptosystem until he has been able to discover in some detail what the basic processes of encryption are. This is borne out by the very considerable investments our own SIGINT organization has made simply to find out how target systems work; it's a prerequisite to any subsequent analysis.