RESEARCH ARTICLE OPEN ACCESS

Secure Communications over Insecure Networks Using Cryptogram

Pritam Singh Patel[1], Ravendra Ratan Singh[2], Satya Patra[3]
Department of Electronical Engineering
Rajashri Shahu College Of Engineering
Tathwade, Pune

Abstract:

It is necessary according to traditional conceptions of cryptographic security, to transmit a key, by secret means, before the encrypted messages can be sent securely to the destination. To select a key over open communications channels is possible in such a fashion that communications security can be maintained is what this paper shows. A method is described which forces any enemy to expend an amount of work which increases as the square of the work required of the two communicants to select the key. Against the passive eavesdropper, the method provides a logically new kind of protection. Both in a theoretical and a practical sense, it suggests that further research on this topic will be highly rewarding.

Keywords — **Source , Network communication, Cryptogram.**

INTRODUCTION:

People have been communicating with each other for many millennia. They often wish to communicate secretly. Since the first use of Caesar's cipher, some two thousand years ago, people have employed a number of ciphers and codes in attempts to keep their correspondence secret. These have met with varying degrees of success until the modem age. The modem digital computer has made it possible to create ciphers which are, in practical terms, unbreakable.1 (At least, if anyone has broken them, they are maintaining a discreet silence.) Underlying this success has been a very definite paradigm, which makes very definite assumptions about the nature of the encryption process, and the conditions under which secret communications can take place. It is the purpose of this paper to consider this paradigm, and to question the assumptions which underlie it. One assumption (that we must transmit a key, by secret means, prior to an attempt to communicate securely) which has traditionally been regarded as a necessary precondition for cryptographically secure communications is not, in fact, necessary. This is demonstrated by exhibiting a solution which allows two communicants to select a key publicly, but in such a fashion that no one else can easily determine it. The body of the paper begins with an explanation of the

traditional paradigm and then develops a new paradigm, which differs significantly from the traditional one. We then argue that the new paradigm is consistent with secret and secure communications. Finally, the implications of the new paradigm are explored in more detail, with the aid of some examples.

Introduction to Cryptography

It is sometimes necessary to communicate over insecure links without exposing one's systems. Cryptography—the art of secret writing—is the usual answer. The most common use of cryptography is, of course, secrecy. A suitably encrypted packet is incomprehensible to attackers. In the context of the Internet, and in particular when protecting wide-area communications, secrecy is often secondary. Instead, we are often interested in the implied authentication provided by cryptography. That is, a packet that is not encrypted with the proper key will not decrypt to anything sensible. This considerably limits the ability of an attacker to inject false messages. Before we discuss some actual uses for cryptography, we present a brief overview of the subject and build our cryptographic toolkit. It is of necessity sketchy; cryptography is a complex subject that cannot be covered fully here. Readers desiring a more complete treatment should consult any of a number of standard references, such as [Kahn, 1967], [Denning, 1982], [Davies and Price, 1989], or [Schneier, 1994]. We next discuss the Kerberos Authentication System, developed at MIT. Apart from its own likely utility—the code is widely available and Kerberos is being considered for adoption as an Internet standard—it makes an excellent case study, since it is a real design, not vapourware, and has been the subject of many papers and talks and a fair amount of experience. Selecting an encryption system is comparatively easy; actually using one is less so. There are myriad choices to be made about exactly where and how it should be installed, with trade-offs in terms of economy, granularity of protection, and impact on existing system.

Secure channels in the real world

There are no perfectly secure channels in the real world. There are, at best, only ways to make insecure channels (e.g., couriers, homing pigeons, diplomatic bags, etc.) less insecure: padlocks (between courier wrists and a briefcase), loyalty tests, security investigations, and guns for courier personnel, diplomatic immunity for diplomatic bags, and so forth.

In 1976, two researchers proposed a key exchange technique (now named after them) — Diffie–Hellman key exchange (D-H). This protocol allows two parties to generate a key only known to them, under the assumption that a certain mathematical problem (e.g., the Diffie–Hellman problem in their proposal) is computationally infeasible (i.e., very very hard) to solve, and that the two parties have access to an authentic channel. In short, that an eavesdropper—conventionally termed 'Eve', who can listen to all messages exchanged by the two parties, but who cannot modify the messages—will not learn the exchanged key. Such a key exchange was impossible with any previously known cryptographic schemes based on symmetric ciphers, because with these schemes it is necessary that the two parties exchange a secret key at some prior time, hence they require a confidential channel at that time which is just what we are attempting to build.

It is important to note that most cryptographic techniques are trivially breakable if keys are not exchanged securely or, if they actually were so exchanged, if those keys become known in some other way — burglary or extortion, for instance. An actually secure channel will not be required if an insecure channel can be used to securely exchange keys, and if burglary, bribery, or threat isn't used. The eternal problem has been and of course remains — even with modern key exchange protocols — how to know when an insecure channel worked securely (or alternatively, and perhaps more importantly, when it did not), and whether anyone has actually been bribed or threatened or simply lost a notebook (or a notebook computer) with key information in it. These are hard problems in the real world and no solutions are known — only expedients, jury rigs, and workarounds.

A New Approach

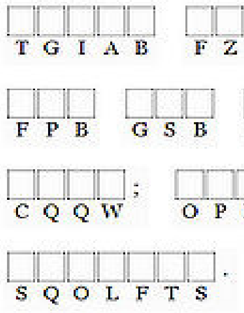
We modify the traditional paradigm by dropping the second restriction on the key channel: that is to say, we no longer demand that Z be unable to determine what is sent on the key channel. Even stronger, we assume that Z has perfect knowledge of everything that is sent over this channel. It is the thesis of this paper that secure communications between X and Y can still take place, even under the highly restrictive conditions we have described. The reader should clearly understand that no key lurks in the background. There is no method by which X and Y can communicate other than the normal channel and the key channel. They have made no secret preparations prior to the time that they wish to communicate securely. We must carefully consider what constitutes a solution. If X and Y eventually agree upon a key, and if the work required of Z to determine the key is much higher than the work put in by either X or Y to select the key, then we have a solution. Note that, in theory at least, Z can determine the key used in most methods simply by trying all possible keys and seeing which one produces a legible message. However, this means that Z must put in an amount of work that is exponentially larger than the amount of work put in by X or Y. The current solution is not exponential. The amount of work required of Z to determine the key will increase as the square of the amount of work put in by X and Y to select the key. Clearly, it would be desirable to find a solution in which the amount of work put in by Z increases exponentially as a function of the amount of work put in by X and Y. We see no reason why such exponential methods should not exist.

Solving a cryptogram

Cryptograms based on substitution ciphers can often be solved by frequency analysis and by recognizing letter patterns in words, such as one letter words, which, in English, can only be "i" or "a" (and sometimes "o"). Double letters, apostrophes, and the fact that no letter can substitute for itself in the cipher also offer clues to the solution. Occasionally, cryptogram puzzle makers will start the solver off with a few letters.

THE METHOD

The method used is based on a single concept: that of a "puzzle." We define a puzzle as a cryptogram which is meant to be broken. A cryptogram is a type of puzzle that consists of a short piece of encrypted text. Generally the cipher used to encrypt the text is simple enough that the cryptogram can be solved by hand. Frequently used are substitution ciphers where each letter is replaced by a different letter or number. To solve the puzzle, one must recover the original lettering. Other types of classical ciphers are sometimes used to create cryptograms.


T G I A B   F Z U   T G P H V G H P B
F P B   G S B   B T T B Z V B   Q R   F
C Q Q W;   O P B F G   K U B F T   F P B
S Q O L F T S.

Example cryptogram. When decoded it reads: "Style and structure are the essence of a book; great ideas are hogwash." -Vladimir Nabokov

A puzzle, though, is meant to be solved, while ideally, a cryptogram cannot be cryptanalyzed. To solve a puzzle, all one need do is put in the required amount of effort.

Based on solving puzzles: small cipher texts designed to be broken. The protocol was originally devised in 1974, and a revised version of the paper published in 1978. The paper starts by talking about the problems with traditional crypto methods, which can be summarised as two points:
Traditional crypto requires a secret key, known only to the legitimate participants.
Traditional crypto assumes the existence of a totally secure channel in order to distribute this key.

The solution is to not have a secure channel! The contribution of this paper is the idea that even when an attacker has perfect information of all the communications, a secure key can still be decided upon by the participants without an attacker being able to easily get it. More precisely, that an attacker would have to put in significantly more work than the participants to determine the key.

In this algorithm, the attacker needs to put in O(N^2) work, whereas each participant only needs O(N) work.
If we call the two participants Alice and Bob, the key decision process goes like this:

1. Alice and Bob agree on some number N.
2. Alice generates N puzzles, where the work required to break a puzzle is O (N). More specifically:
   - A puzzle is an encrypted string consisting of a random ID number, a random key, and some constant string.
   - Encryption is done by using some strong algorithm and restricting the size of the key space to some linear function of N.
   - Each puzzle is encrypted with a different random key from this key space (note that this is not the same as the key included in the puzzle clear text).
3. Alice transmits all the puzzles to Bob.
4. Bob picks one puzzle at random, and solves it. Specifically:
   - Bob brute-forces the key of the puzzle (this is the only possible method, as a strong encryption function was chosen). Bob can check that a puzzle was correctly decrypted by checking for the agreed-upon constant string.
5. Bob transmits the ID number of the chosen puzzle to Alice.
6. Alice and Bob now use the key from that puzzle for all further communications.

Let’s introduce an attacker Eve, and summarise what they all know after this exchange:

Alice knows the N puzzles; the clear text of all puzzles, Bob’s chosen ID number, and the corresponding key.

Bob knows the N puzzles, the clear text of one puzzle, the ID number, and the corresponding key.

Eve knows the N puzzles and the ID number.

The only way for Eve to get the corresponding key is to solve puzzles at random until she finds one with a matching ID number. This will require solving N/2 puzzles on average, which corresponds to O(N^2) time, as each puzzle takes O(N) to solve.

Using these conventions, we can write the algorithm for Alice, who is generating the puzzles, in the following fashion:

```
var ID. KEY. CONSTANT. RANDOMKEY,
PUZZLE. K I. K2:bit string;
begin
K I:=RAND(LARGE);
```

```
K2:=RAND( LARGE);
CONSTANT:=RAND( LARGE);
TRANSMIT( CONSTANT);
for 1:=1 to N do
begin
ID:=F(Kl,I);
KEY:=F( K2,ID);
RANDOMKEY:<=&RAND( C*N);
end;
PUZZLE:=F( RANDOMKEY. lD. KEY. CONSTANT);
TRANSMIT( PUZZLE);
end;

end;
```

Merkle goes further than just proposing a key exchange algorithm, he anticipates the development of publicly-known keys and keyservers! He discusses this in the context of an organisation wishing to have private communication in the face of an enemy, based on codebooks:

First, each unit or command that wished to be in the code book would generate its own first transmission [the constant string and the N puzzles]. These would all be sent to a central site, where the names and first transmissions of all involved communicants would be entered into the code book. The codebook would then be distributed. In essence, we are simply specifying the nature of the communication channel between X and Y. It is not a direct communication channel, but is somewhat roundabout. X publishes his first transmission in the codebook, along with his name. The return transmission from Y to X can now take place over normal communication channels. Y is assured that he is talking to X, because Y looked up X’s first transmission in the codebook. At this point X and Y have established a common key, but X does not know that he is talking to Y. Anyone could have sent the return transmission, claiming they were Y. To avoid this, X and Y repeat the process of selecting a key, but X now looks up Y in the codebook, and sends a return transmisison to Y, based on Y’s first transmission. The return transmission will be meaningful only to Y, because the return transmission is based on Y’s first transmission. X knows Y’s first transmission came from Y, because it is entered in the codebook. If X and Y now use both keys, then they are assured they are talking to each other, and no one else. To summarize: using only a codebook, which is assumed to be correct, but which is not assumed to be secret, X and Y have established an authenticated, secure communications channel. They have done so quickly and easily. The key need be used for only a short period of time (a single conversation), and can then be changed with equal ease.

A more familiar discussion then follows proposing effectively the same protocol, but in the context of computer systems. The compiler of the codebook is the network administrator, and the codebook is the listing of users.

It would be no exaggeration to say that, without this contribution, public-key cryptography would have been much slower to develop, and the state of secure communication would not be as happy as it is today. Furthermore, like many papers introducing an entirely new field, this one is *simple*, it’s *easy to read*, and it doesn’t require a lot of background knowledge. The algorithm described can be implemented in a few dozen lines of code.

At the very end, X must receive the ID that Y transmitted, and deduce the key. The last actions that X must perform are as follows:

```
begin
RECE'VE( 10):
KEV:=P( K2,1D);
comment KEY now has the same value in both X and Y. All they
have to do is use KEY as the key with which to encrypt further
transmissions.
end:
```

The only information available to Z is the code executed by X and Y, and the values actually transmitted over the key channel. Thus, Z is in possession of N, the CONSTANT, the ID that Y transmitted to X, and also the puzzles that X transmitted to Y. All other variables are known either exclusively by X, or exclusively by Y. In summary: the method allows the use of channels satisfying assumption I, and not satisfying assumption 2, for the transmission of key information. We need only guarantee that messages are unmodified, and we no longer require that they be unread. If the two communicants, X and Y, put in *O(N)* effort, then the third person, Z, must put in *O(* Nf2) effort to determine the key. We now tum to the consideration of various implications of this work.

Thoughts

Here are some discussion points if you want to talk about this paper with others:

- The paper mentions an attacker discovering a secret key being transmitted over a secure channel (as in traditional crypto) by “practical cryptanalysis", a euphemism for physically intercepting the message. Public-key crypto solves this to some extent, but is the issue of "practical cryptanalysis" totally solved?

- This algorithm requires an attacker to put in O(N^2) work to determine the key, whereas the communicants only need O(N) work. Quadratic time isn't generally regarded as being very good for crypto nowadays. Why?
- As all the communication is public, an attacker could just record everything said and gain access to all communications past, present, and future when they eventually crack the key. Why isn't this a huge flaw with public-key cryptography?

A key distribution system based on the current ideas might proceed as follows. First, each unit or command that wished to be in the code book would generate its own first transmission. These would all be sent to a central site, where the names and first transmissions of all involved communicants would be entered into the code book. The codebook would then be distributed. In essence, we are simply specifying the nature of the communication channel between X and Y. It is not a direct communication channel, but is somewhat roundabout. X publishes his first transmission in the codebook, along with his name. The return transmission from Y to X can now take place over normal communication channels. Y is assured that he is talking to X, because Y looked up X's first transmission in the codebook. At this point X and Y have established a common key, but X does not know that he is talking to Y. Anyone could have sent the return transmission, claiming they were Y. To avoid this, X and Y repeat the process of selecting a key, but X now looks up Y in the codebook, and sends a return transmission to Y, based on V's first transmission. The return transmission will be meaningful only to Y, because the return transmission is based on V's first transmission. X knows Y's first transmission came from Y, because it is entered in the codebook. If X and Y now use both keys, then they are assured they are talking to each other, and no one else. To summarize: using only a codebook, which is assumed to be correct, but which is not assumed to be secret, X and Y have established an authenticated, secure communications channel. They have done so quickly and easily. The key need be used for only a short period of time (a single conversation), and can then be changed with equal ease. The new paradigm also has implications for network security. In a computer network, with many users with diverse needs, security is difficult to maintain. If the codebook in the previous example were compiled at the same time and by the same people who normally compile the directory of network users, the additional effort required would be minimal. Those network users interested in security would submit a first transmission to be included next to their entry in the network directory. They would also make sure that their copy of the network directory was correct. Those users not interested in security would ignore the security procedures. Diverse needs, ranging from no security, to very tight security, could then be met on the same network. As a final example, consider the following situation. Assume two forces, Us and Them, are fighting. They are winning, because they have broken our codes and ciphers. We only find out about this when we discover that they attack exactly where we are weakest, retreat just before our attacks, and generally seem to know too much too quickly. Our forces are in the field, fully deployed, with no chance for distributing new keys in accordance with the traditional paradigm. Under the traditional paradigm, we are lost. Using the new paradigm, we can easily change all our keys, and re-establish security. The difference is dramatic.

We summarize the discussion to the current point. The traditional paradigm for cryptographically secure communications was examined. A new paradigm was proposed, and a method of key distribution was described which is consistent with the new paradigm. The only weakness in the method is that it is O(Nf2), and not exponential. The weaker restrictions on the key channel demanded by the new paradigm open up the possibility of using more normal, i.e., cheaper, channels of communication with which to update keys. In addition, violation of the weaker restriction on the key channel can be detected and corrective action taken. Violation of the stronger restriction that the key channel must be unreadable might go unnoticed, and result in catastrophic loss of security. This possibility is eliminated with the new paradigm. In the event that there is no channel available which satisfies the stronger restriction, but there is a channel which satisfies the weaker restriction, then the current method provides an option which is otherwise unavailable.

CONCLUSION

We discussed about cryptography, secure channels, the Puzzle Method and all. Then we moved to the secret key, O (Nf2) is the method this paper dealt with. If an exponential method were possible, it would offer such significant advantages that it would almost surely supplant them in short order over traditional techniques. The problem appears to often enough leverage that it can be attacked, an exponential solution would offer significant practical advantages and as witness the current solution, over traditional techniques. The problem merits serious consideration.

REFERENCES

1. Diffie. W .• and Hellman. M. New directions in cryptography. *IEEE Trans. on Inform. IT-22.* 6 (Nov. 1976).644-654.

2. Feistel. H. Cryptography and computer privacy. *Sci Amer.* 228.5 (May 1973). 15-23.

3. Kaba. D. *The Codebreakers.* MacMillan. New York, 1976.

4. Merkle. R .• and Hellman. M. Hiding information and receipts in trap door knapsacks. To appear. IEEE Trans. on Inform.

5. Rivest. R.L .• Shamir. A .• and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM* 21.2 (Feb. 1978), 120-126.

6. Shannon. C.E. Communication theory of secrecy systems. *Bell Syst. Tech. J.* 28 (1949).654-715.

7. Wyner. A.D. The wire tap channel. *Bell Syst. Tech. J.* 54. 8 (Oct. 1975). 1355-1387.

8. Michael Walker's Paper : Secure Communications Over Insecure Channels Review of (Merkle 1978), one of the first public-key cryptosystems. Published on October 4, 2015

9. Ran Canetti: Universally Composable Signatures, Certification, and Authentication. CSFW 2004, http://eprint.iacr.org/2003/239

10. Waka Nagao, Yoshifumi Manabe, Tatsuaki Okamoto: A Universally Composable Secure Channel Based on the KEM-DEM Framework. TCC 2005: 426-444