Clarkson University

Computer Science Technical Report

Uncoercible Communication

Josh Benaloh & Dwight Tuinstra

Clarkson University TR-MCS-94-1
March 1994

Uncoercible Communication

Josh Benaloh and Dwight Tuinstra
Clarkson University

Abstract

This paper describes a model and method whereby one agent can send a private message to another over a public channel from within a "hostile" environment in which the sending agent may be subject to extreme coercion both before and after the sending of the message. Coercive forces may demand that certain information be or not be sent, may monitor the channel over which the transmission will take place, and may require that the sending agent reveal all information after the transmission is complete. Nevertheless, the sending agent may claim to have sent one message while actually having sent another and will be *unable* to provide any kind of receipt to the coercer to show what message was actually sent.

1 Introduction

A central issue in many cryptographic protocols is *secrecy* — intuitively, the property that when a plaintext M is encrypted to ciphertext $C = E(M)$, only the intended recipient(s) can decrypt the message. In public-key cryptosystems, the sender cannot (necessarily) decrypt the message; however, the sender can still prove that a given message was sent. This is done by displaying a plaintext M, encryption parameters, and any other values generated during encryption which, when applied to the message M, yield a ciphertext which matches the (publicly-observable) ciphertext C.

This is adequate for situations in which the sender cannot be coerced into revealing the message. However, there are situations where *uncoercibility* is also desirable — intuitively, the property that the sender cannot prove the contents of a given message (and hence cannot be made to divulge the message).

An election system in which voters can reveal the value of their votes is open to fraud. "Ward bosses" can reward voters who show they voted in a certain way and threaten reprisals against voters who voted "improperly" or who refuse to divulge their votes. Uncoercibility removes such pressures — if voters cannot prove the "value" of their votes, they cannot be punished for voting "improperly" nor be rewarded for voting "properly".

One can imagine other situations in which uncoercibility is desirable. A government auditor investigating corruption and mismanagement may wish to have encrypted affidavits from

employees of, say, a nuclear power plant, defense contractor, or intelligence agency. Employees must not be subjected to coercion from supervisors if the auditor is to gain significant information from such inquiry. In the cloak-and-dagger realm, an agency headquarters would like to send out field agents who report back using an uncoercible cryptosystem. Then, short of being under direct observation while enciphering a message, agents would be able to indicate whether a message is "genuine" or is being sent under coercion.

This last scenario reveals a critical premise for uncoercibility: the sender of the message, when actually encrypting the message, must not be under direct supervision or observation of the coercing entity. That is, there must be a physical separation. This is not as unrealistic or burdensome as it seems. We have come to accept and expect the privacy of the voting booth. An auditor might provide secure facilities for the private composition of messages. Double agents must be able to move about and act as if they have not been compromised.

Uncoercibility is achieved by making coercion pointless — although coercion may constrain the set of the sender's possible messages, the protocols here will, with overwhelming probability, allow the sender to easily construct a message which warns the recipient that coercion is taking place. Furthermore, this warning message is (to the coercer), *perfectly* indistinguishable from the message the coercer wants sent.

1.1 Related Work

In [BeTu94], the idea of uncoercible communication is introduced in the narrow context of secret-ballot elections. That context is very limited since a voter's list of choices in an election is small, and the constraints are different since the voting process requires a form of external verifiability.

Perhaps the closest prior work to uncoercible communication is the body of work exploring subliminal channels and the abuses they can allow. This idea was introduced in [Simm83] and methods of both exploiting and preventing these abuses are described in work including [Simm84], [Simm85], [Desm88a], [Desm88b], and [Desm89]. The problem examined in this body of prior work is that of a sender transmitting a message to a recipient in the presence of an observer who is to be allowed to restrict the contents of the message.

The problem of uncoercible communication is, in some sense, complementary to that of abuse-free channels. Here, the observer has access to the recipient before and after the transmission of a message and can also observe the transmission. Nevertheless, it should not be possible for the observer to determine or influence the contents of the message.

1.2 Organization

Section 2 motivates and develops the model, and provides the formal definition of uncoercibility. Section 3 describes two protocols for achieving uncoercible communication. The proof of uncoercibility follows in section 4. In section 5 we discuss extensions and practical matters relating to the protocols.

2 The Model

The model in which uncoercible communication can be achieved is admittedly somewhat strange at first glance and requires motivation. We assume the existence of a one-way private channel *from* the message recipient *to* the message sender and a public channel which can be used to send information from the message sender to the message recipient. Although the notion of a one-way private channel may seem contrived, a weaker assumption actually suffices for this work. It is sufficient for the sender and recipient to both have (read-only) access to a shared private stream of random bits. This can, for instance, easily be achieved by equipping the sender and recipient with identical physical devices which produce elements of a cryptographically secure pseudo-random sequence at fixed intervals. (Such devices are currently marketed and widely used in lieu of passwords for account access.) These devices can be used as an effective one-way recipient-to-sender channel, and the methods of this paper will show how such devices can be used to ensure uncoercible communication.

It is also necessary for the sending agent to be physically separated from any coercive forces *at the time* the message is being sent. However, the sending agent may be subject to coercion immediately before and immediately after transmitting a message, and the public channel used to send the message may be monitored by coercive forces. It is evident that without the physical separation, a coercive agent could effectively replace the sender and, by extracting information from and giving instructions to the sender, could send a message of its choosing.

If a private channel from the sender to the recipient existed, the problem would become trivial since the sender could simply send any desired message over this channel and, with no receipt available, could later claim that a different message was sent. However, the availability of a private channel from the sender to the recipient is unrealistic in many circumstances.

It should be noted that the existence of a one-way private channel from the recipient to the sender *does not* trivialize the problem. It is not immediately apparent how a channel in the seemingly "wrong" direction can be exploited to achieve uncoercibility. We show how to exploit such a channel, and demonstrate that in practical terms, a serial protocol (in which transmissions on the two channels are interleaved) is preferable to a parallel protocol (in which all communication on the private channel occurs before any on the public channel).

It is not immediately apparent whether or not the "obvious" solution, of having the recipient use the private channel to transmit a one-time pad, truly solves the problem. Perhaps, before the transmission, the coercer could supply the sender with a one-way function f ([DiHe76]) which is to be applied to the key K, and which dictates the observable values of some (or all) of the transmitted bits T. After the transmission has occurred, the coercer could demand a copy of K and check to see if T indeed conforms to the constraints given by $f(K)$. If such a function incorporates a message the *coercer* wants sent, to what extent is it possible for the sender to modify the message, or warn the recipient that the message is being sent under coercion?

We provide answers to these question, showing that while a conventional one-time pad

system can be made to work when the number of meaningful messages is an exponentially small fraction of all messages, it will not work in the general situation. The protocols we present provide uncoercibility regardless of the size of the message space or the size of the subspace of meaningful messages. This is highly desirable, for example, in situations where a "commit/no commit" message is to be sent.

Formally, we say that a *communication system* is a pair of protocols $\mathcal{S}$ and $\mathcal{R}$ to be executed by a sender and a recipient. $\mathcal{S}$ takes a message M and a private channel data stream K (provided by — or simply shared with — the recipient) as input and produces as outputs a transmission stream T and a possible private record W which may be demanded by the coercer as a receipt. The protocol $\mathcal{R}$ takes a transmission stream T and its own data stream K as input and produces two outputs: a message M and a single acceptance bit D. When the communication system is run with $\mathcal{S}$ on message M and any private data stream K provided by $\mathcal{R}$, the transmission T it outputs should be such that $\mathcal{R}(T, K)$ yields (M,*true*). The interpretation of D is that if D is *true* then the message M is to be accepted as uncoerced, but if D is *false* then M is to be regarded as a coerced message.

A communication system is said to be *uncoercible* if, for any substitute sender protocol $\tilde{\mathcal{S}}$, there is an additional protocol $\tilde{\mathcal{S}}'$ which is easily computable from $\tilde{\mathcal{S}}$ and which satisfies the following: the distribution of output pairs (T, W) produced by runs of the communication system in which protocol $\tilde{\mathcal{S}}$ is executed is identical to the distribution of output pairs (T', W') produced by runs in which protocol $\tilde{\mathcal{S}}'$ is executed (perfect indistinguishability); and applying $\mathcal{R}$ to a transmission T' produced by $\tilde{\mathcal{S}}'$ will (with overwhelming probability) yield an acceptance bit D of *false*. The distributions here are taken over the possible values of the private data stream K, and overwhelming probability here means probability at least $1 - 2^{-N}$ where N is a predetermined security parameter.

Intuitively, this definition allows a sender who has been instructed by a coercer to use protocol $\tilde{\mathcal{S}}$ rather than $\mathcal{S}$ to instead use protocol $\tilde{\mathcal{S}}'$ which, to the coercer, is indistinguishable from $\tilde{\mathcal{S}}$ and which, with overwhelming probability, alerts the recipient to the coercion.

3 The Method

In what follows, we use the notational conventions hinted at above: a tilde indicates a protocol or message which a coercer is demanding be sent (or used); and a prime mark indicates a protocol or message which the sender uses in countering the coercion.

We also assume, as part of the protocols, that a security parameter N and a block-length constant L have been published by the recipient.

As a prelude to developing the methods we ask: what is the nature of the coercer's power? As noted in the model, the coercer can demand that a substitute protocol $\tilde{\mathcal{S}}$ be run in place of the sender's protocol $\mathcal{S}$, and can demand a receipt W after the transmission is complete. The substitute protocol can be viewed as a *coercive function* f which takes as argument the key K (which is not observable to the coercer) and which produces the transmission $T = f(K)$ and, as a side effect, a receipt W.

The coercive function f can be viewed as completely deterministic. There is no need for f to be randomized, since a randomized function can be though of as a deterministic function with an implicit random tape. Since there is no advantage to the coercer to allowing the sender (or anyone else) to supply the tape, the random tape might just as well have been provided by the coercer — but then, the function might just as well be a deterministic function provided by the coercer.

There is also no advantage to the coercer in allowing the sender any leeway in choosing transmission bits. Any instructions that the coercer might give to the sender can be viewed as being encoded in the function f (this includes the actual message to send, or linguistic instructions such as "after the function gives you the basic message, adjust word endings to be grammatically correct"). In making this assumption, we are perhaps allowing the coercer to define a function beyond the bounds of current technology, but note that we are *strengthening* the hand of the coercer. Even with this assumption, however, the protocols do not suffer.

As for the receipt W, it suffices for the coercer to demand the one-time pad read from the private channel. Other items, such as the intermediate results of computations, carry no more information than the key — if the sender is able to "forge" a false but acceptable key as a receipt, then the sender is able to forge any intermediate results based upon that key. This is done by simply discarding whatever computations generated the forged key K', and applying f to K' as though it had been received on the private channel.

The sum total of what a coercer can do is therefore: (1) provide the sender with a fixed function f dependent on the one-time pad K; (2) monitor the public channel over which the transmission is sent to the recipient; and (3) demand that the entire one-time pad K be revealed after the conclusion of the transmission.

Thus, when a transmission is finished, the coercer has a copy of the transmission T and the receipt W (which the sender is implicitly claiming is the key K). The coercer, to see if the sender followed instructions, is limited to checking whether $f(W) = T$.

3.1 One-Time Pads Used in Parallel

We now turn to the protocols for the case in which keys are transmitted in parallel to the sender (that is, the sender has access to the entire key before beginning the transmission of the ciphertext on the public channel).

The sender follows protocol $\mathcal{S}$: first, read K from the private channel. If not under coercion, the message M (of length L) is sent by setting the first L bits of T equal to the (bitwise) XOR of M with the first L bits of K, and setting the last N bits of T equal to the last N bits of K. That is, the message is sent in the normal fashion for a one-time pad, followed by a verification tag in which the sender transmits the last N bits of the key in cleartext.

The recipient's protocol $\mathcal{R}$ is as follows: transmit a key K, of length $L+N$, on the private channel. Once this is complete, monitor the publicly-observable channel for the transmission T, of the same length. If the last N bits of T (the verification tag) are equal to the last N

bits of K, set the acceptance bit D to *true* and form the message M by (bitwise) XOR-ing the first L bits of K with the first L bits of T. In all other situations set the acceptance bit D to *false*.

If the sender *is* under coercion, (that is, has been provided with a coercive function f), the sender also reads K from the private channel, but additionally constructs, independent from K, a second key K' which *could have been* transmitted by the recipient (perhaps by using the same pseudo-random number generator as the recipient, or by reading another key from the private channel). The sender then transmits $T = f(K')$ and retains K' as a receipt.

In section 4.1, we show that this protocol describes an uncoercible communication system.

3.2 One-Time Pads Used in Serial

Although it does not seem to offer any theoretical advantages, there may be substantial practical advantages to establishing protocols in which the private pad bit stream K is received by the sender in a serial fashion. These advantages include (once again in practice rather than theory) a possible obviation of the need to require a verification tag and an ability to send uncoerced information while being coerced. These added capabilities will be discussed in section 5.

By a *serial* system, we mean a system in which the one-time pad K is transmitted to the sender one bit at a time. Once a bit of the key has been sent, the recipient waits for a transmission bit before sending the next key bit.

The nature of the serial protocol demands that the function f be defined slightly differently. In particular, it takes the key bits $k_1, k_2, \ldots, k_i$ that have been *seen so far* and outputs the bit t_i to be transmitted. However, an assumption that it is deterministic, and dependent only on the key, is still a *strengthening* assumption for the coercer. In particular, we do not need to assume it is dependent on any bits of the transmission.

As in the parallel case, the extent of the coercer's power is to provide the sender with a coercion function f, to monitor the public channel to record the transmission T, and to demand the pad as receipt W. Also as in the parallel case, when checking to see if the sender followed the coercer's instructions, the coercer is limited to comparing the transmitted stream T with what the function f would have transmitted if given the (claimed) pad W.

The recipient's protocol $\mathcal{R}$ is straightforward: successively send $L + N$ key bits k_i on the private channel, waiting for the transmission bit t_i to appear on the public channel before sending the next key bit. If the last N transmission bits are equal to the last N key bits, set the acceptance bit D to *true* and set the message M to the bitwise XOR of the first L bits of the key with the first L transmission bits. In all other situations set D to *false* .

If no coercion is present, the sender's protocol $\mathcal{S}$ is exactly as expected. For the first L bits of the key, transmit the XOR of that bit with the respective bit of the message: $t_i = k_i \oplus m_i$. For the remaining N bits of the transmission, send the key bits unchanged: $t_i = k_i$.

If there is coercion present, the sender takes advantage of a simple yet interesting property of the coercion function f. Recall that f completely determines the transmission bits that

are to appear; thus the transmission bits are given by $t_i = f(k_1, k_2, \ldots, k_i)$. We say that a transmission bit t_i is *forced* if $f(k_1, k_2, \ldots, k_i) = f(k_1, k_2, \ldots, \overline{k_i})$, and we say t_i is *free* if $f(k_1, k_2, \ldots, k_i) \neq f(k_1, k_2, \ldots, \overline{k_i})$. For the first L bits the sender records the receipt bit $w_i = k_i$ and transmits $t_i = f(w_1, w_2, \ldots, w_i)$. During the remaining N bits the sender continues this procedure for as long as the t_i are forced. If all of the last N bits of the transmission are forced, this completes the protocol. However, if a free bit should occur, on the first such occurrence (say bit n), the sender transmits the complement of the key $t_n = \overline{k_n}$. If $f(w_1, w_2, \ldots, w_{n-1}, k_n) = \overline{k_n}$ the sender records the receipt bit $w_n = k_n$, otherwise the sender records the receipt bit $w_n = \overline{k_n}$. After this bit has been sent the sender continues as before, recording $w_i = k_i$ for the receipt and sending $t_i = f(w_i, w_2, \ldots, w_n, \ldots, w_i)$ for the remainder of the transmission.

In section 4 below, we prove that this protocol results in an exponentially small probability of an undetected coercion.

4 Uncoercibility

It is not at all difficult to see that the protocols described in section 3 form uncoercible communication systems.

First, in both parallel and serial cases, the fact that the protocols given describe a communication system follows directly from the properties of one-time pads. In particular, the only property utilized for a communication system is the fact that XORing a message with a pad twice returns the message.

To show that these protocols form an *uncoercible* communication system, it will be necessary to show how any coercive sender protocol $\tilde{\mathcal{S}}$ can be adapted to form a protocol $\tilde{\mathcal{S}}'$ which, to the coercer, is perfectly indistinguishable from $\tilde{\mathcal{S}}$ and yet causes the recipient to produce an acceptance bit D of *false*.

4.1 Uncoercibility of the Parallel Protocol

In the parallel protocols, as was argued in section 3.1, the transmission stream T produced by $\tilde{\mathcal{S}}$ can be viewed as a deterministic function f of the pad K.

As described in section 3.1, $\tilde{\mathcal{S}}'$ is defined to behave like $\tilde{\mathcal{S}}$, with the exception that $\tilde{\mathcal{S}}'$ is given a different key K' than the key K given to $\tilde{\mathcal{S}}$, where K' is chosen randomly and independently of K.

Since the key K is random, the substitute key K' claimed by $\tilde{\mathcal{S}}'$ is just as likely to have been the actual key as K. Furthermore, the observed transmission stream T is generated by applying f to K'. A receipt record W containing $(f(K'), K')$ is just as plausible as the receipt record containing $(f(K), K)$ that would have been produced by $\tilde{\mathcal{S}}$. The distribution of output pairs (T, W) produced by runs of $\tilde{\mathcal{S}}'$ is thus completely indistinguishable from the distribution of output pairs produced by runs of $\tilde{\mathcal{S}}$.

It is not hard to see that when $\widetilde{\mathcal{S}}'$ is run, the transmission $f(K')$ is rendered into random bits when XOR-ed with K by the recipient. The probability that all N bits of the verification tag are equal to the respective N bits of K (and the verification bit D returned by $\mathcal{R}$ is thus set to *true*) is 2^{-N}. Hence, the communication system of section 3.1 is uncoercible.

4.2 Uncoercibility of the Serial Protocol

In the serial protocols, as was argued in section 3.2, the transmission stream T produced by $\widetilde{\mathcal{S}}$ can be viewed as a deterministic function f of the bits of the pad stream K seen so far; that is, the transmission bit t_i is given by $t_i = f(k_1, k_2, \ldots, k_i)$.

As described in section 3.2, $\widetilde{\mathcal{S}}'$ is defined to behave exactly like $\widetilde{\mathcal{S}}$ with one exception: if any free bits occur in the verification tag, at the first such occurrence (say, at the n^{th} bit of the key), $\widetilde{\mathcal{S}}'$ sets the value of the transmission bit t_n to the complement of the key bit k_n; and records as a receipt bit w_n the value which, if it *were* the key bit, would cause f to generate the value of t_n. Since t_n is part of the verification tag, the recipient expects to see $t_n = k_n$ and will set its acceptance bit D to *false* when instead $t_n = \overline{k_n}$ is returned.

By the construction of the altered pad stream K', f applied to K' produces the observed transmission stream T (this, indeed, is exactly how T was generated). Since the pad stream K is random, the altered pad stream K' claimed by $\widetilde{\mathcal{S}}'$ is just as likely to have been the actual pad as K. Thus, the receipt record W produced by $\widetilde{\mathcal{S}}'$ containing K' rather than K and the associated transmission stream T are just as plausible as the record that would have been produced by $\widetilde{\mathcal{S}}$, so the coercer can see nothing amiss.

The coercer's only hope is therefore to not allow *any* of the bits of the verification tag to become free. However, making all N bits of the verification tag forced means that each of the N values to be transmitted must have been decided upon *before* the associated pad bit values are seen! Since these pad bits are random, and since the tag's bit values must match these N random bits, the chance of the coercer avoiding detection is at most 2^{-N}. The communication system of section 3.2 is therefore uncoercible.

5 Discussion and Open Problems

In practice, a truly random sequence may be difficult (if even possible) to produce. Cryptographically secure pseudo-random sequences are quite sufficient here, but the uncoercibility of the communication system becomes dependent on the cryptographic assumption(s) of the underlying pseudo-random generator. It is easy to see how a coercer who can determine the pad bits in advance can force a proper verification tag to be produced. However, the ability to determine whether a bit sequence is or is not producible by the recipient's pseudo-random number generator would give a coercer an advantage in determining (after the fact) whether or not its instructions were followed.

In the serial system of section 3.2, we have required that an N bit verification tag be sent at the *end* of the transmission. This is not an essential requirement. One can imagine

an uncoercible system in which the odd-numbered bits are considered verification bits, and even-numbered bits are considered the message. The system achieves the desired level of security as long as any N bits are reserved for the verification tag.

A weakness of the systems presented here is their "all or nothing" nature — the recipient accepts or rejects the message in total. One can envision situations where this is undesirable. For example, a sender may have only one chance to send critical information, yet be under coercion.

In the parallel case, the sender is limited to the choices of sending the coercer's message or sending an essentially random message which informs the recipient of the presence of coercion. The serial case, however, has the practical advantage that the sender may be able to take advantage of the bits which the coercion function leaves free. By judicious choice of free bit values (and recording of associated receipt bit values), the sender may be able to construct a message which may be partially garbled but which nonetheless conveys some information of value to the recipient. Since the verification tag occurs at the end of the transmission, the sender has the option to tell the recipient to either accept or reject the transmission. The "all or nothing" problem can be further alleviated by transmitting several smaller (separately verified) messages rather than one large message.

The issue of attempting to send information while under coercion reveals a rather nice symmetry within the serial system. The coercer is in a quandary: any forced bit transmits no information to the recipient (since it is XOR-ed with the random key); yet on any free bit the sender can, with impunity, disregard the value "dictated" by the coercion function and send whatever message bit is desired. That is to say, *the sender can send exactly as much information as the coercer.*

This points out a possible modification to the cryptosystem: if the *meaningful* messages are a small fraction ε of the message space, the verification tag can be omitted. Any meaningful message has at most an ε probability of being coerced. This is true for both serial and parallel systems. Additionally, in serial systems any meaningful *portion* of the message has a low probability of being coerced. However, such systems may no longer meet the definition of uncoercibility: a claim that such a system is uncoercible implicitly assumes that the ratio of meaningful to possible messages is bounded above by $1/2^N$. This is not in general the case (for example, no meaningful message or message portion less than N bits long can be considered uncoerced), and it may be problematical to show this bound for some message spaces. Thus the "obvious solution" mentioned in section 2, of direct use of a one-time pad, may work in some situations but cannot in general be considered to provide uncoercible communication.

We have assumed the existence of a secure one-way private channel from the recipient to the sender for the purpose of sending the bits of the key. As noted in section 2, a weaker assumption actually suffices, namely, for the sender and recipient to have read-only access to a stream of random bits denied to the coercer. In practice, such a stream can be provided by a pair of (synchronized, identical) devices each producing a cryptographically secure sequence of pseudo-random bits. One open question is that of whether or not uncoercible communication can occur when shared access to a secure random bit stream is removed from the model.

The idea behind uncoercibility is that any coercion is sure to be detected by the recipient and is thereby deterred. The price is that the message may be garbled. We can envision a stronger form of uncoercibility, *perfect uncoercibility*, in which coercion is deterred by making it absolutely impossible: the sender can send any message desired via a (cryptographically secure) public transmission T, but can *never* prove anything about the message beyond what is publicly known about T. This seems achievable in a model where there is no communication between coercer and sender prior to the transmission — there is no chance for the coercer to require the sender to retain a receipt. It is, of course, also possible if there is a private channel available *from* the sender *to* the recipient. It is an open question whether perfect uncoercibility is achievable in a reasonable model which allows prior communication between coercer and sender.

References

[BeTu94] **Benaloh, J.** and **Tuinstra, D.** "Receipt-Free Secret-Ballot Elections." To appear in *Proc.* 26^{th} *ACM Symp. on Theory of Computing*, Montéal, PQ (May 1994).

[Desm88a] **Desmedt, Y.** "Subliminal-Free Authentication and Signature." *EuroCrypt '88*, Davos, Switzerland (May 1988), 23–33.

[Desm88b] **Desmedt, Y.** "Abuses in Cryptography and How to Fight Them." *Crypto '88*, Santa Barbara, CA (Aug. 1988), 375–389.

[Desm89] **Desmedt, Y.** "Making Conditionally Secure Cryptosystems Unconditionally Abuse-Free in a General Context." *Crypto '89*, Santa Barbara, CA (Aug. 1989), 6–16.

[DiHe76] **Diffie, W.** and **Hellman, M.** "New Directions in Cryptography." *IEEE Trans. on Information Theory 22*, 6, (Nov. 1976), 644–654.

[Simm83] **Simmons, G.** "The Prisoners' Problem and the Subliminal Channel." *Crypto '83*, Santa Barbara, CA (Aug. 1983), 51–67.

[Simm84] **Simmons, G.** "The Subliminal Channel and Digital Signatures." *EuroCrypt '84*, Paris, France (Apr. 1984), 364–378.

[Simm85] **Simmons, G.** "A Secure Subliminal Channel (?)." *Crypto '85*, Santa Barbara, CA (Aug. 1985), 33–41.