Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack

Charles Rackoff
Daniel R. Simon
Dept. of Computer Science
University of Toronto
Toronto, Ontario M5S 1A4
Canada
rackoff@theory.toronto.edu
me@theory.toronto.edu

Abstract

The zero-knowledge proof of knowledge, first defined by Fiat, Fiege and Shamir, was used by Galil, Haber and Yung as a means of constructing (out of a trapdoor function) an interactive public-key cryptosystem provably secure against chosen ciphertext attack. We introduce a revised setting which permits the definition of a non-interactive analogue, the non-interactive zero-knowledge proof of knowledge, and show how it may be constructed in that setting from a non-interactive zero-knowledge proof system for NP (of the type introduced by Blum, Feldman and Micali). We give a formalization of chosen ciphertext attack in our model which is stronger than the "lunchtime attack" considered by Naor and Yung, and prove a non-interactive public-key cryptosystem based on non-interactive zero-knowledge proof of knowledge to be secure against it.

1 Introduction

A fundamental goal of modern cryptography is to formalize and solve the problem of how a group of parties may communicate securely with each other in an environment which does not necessarily prevent adversaries from eavesdropping on, or tampering with, message transmissions. One very strong (but not necessarily sufficient) criterion for prospective solutions has been security of each party's received messages against "chosen ciphertext attack", described informally as an attack on a particular message during which the attacker may obtain decryptions of other messages chosen at will.

Galil, Haber and Yung ([GHY]) showed formally how zero-knowledge interactive proofs of knowledge could be used to make public-key cryptosystems secure against chosen ciphertext attack. The formal argument follows the intuition of Blum, Feldman and Micali ([BFM]); a legal encryption is defined as including an interactive proof that the sender knows its decryption. Thus a machine which can carry out a successful chosen ciphertext attack can be "interrogated" so as to answer its own requests for decryption. It

J. Feigenbaum (Ed.): Advances in Cryptology - CRYPTO '91, LNCS 576, pp. 433-444, 1992.

can therefore be used by a "blind" attacker (one without the ability to obtain decryptions of chosen ciphertexts) to break the underlying cryptosystem.

Unfortunately, attempts to remove the interaction from the above system face two closely related problems. Firstly, it is not intuitively clear how one could conceivably prove knowledge non-interactively; such a proof would necessarily not be feasibly produceable by parties not possessing the knowledge in question. However, merely receiving a non-interactive proof of knowledge from someone may well allow one to "produce" the same proof of knowledge to someone else. Secondly, the attacker in a chosen ciphertext attack is in general entitled to exploit this difficulty in proving knowledge by including the attacked ciphertext among the chosen ciphertexts to be decrypted, in a so-called "playback" attack.

Naor and Yung ([NY]) solved the latter problem by using a more restricted form of chosen ciphertext attack, known informally as the "lunchtime" or "midnight" attack. In this form of attack, the attacker is provided with decryptions of any messages it can generate, but only *before* receiving the message it is to attempt to decrypt itself. Thus the playback attack problem disappears. In their solution, each message is encrypted using two independent cryptosystems, and both encryptions are sent along with a non-interactive zero-knowledge proof that the two encryptions encrypt the same message. It turns out that an attacker capable of decrypting messages thus encrypted (when permitted a lunchtime chosen-ciphertext attack) can be used to decrypt messages in an unknown cryptosystem.

However, the "lunchtime" chosen ciphertext attack, being weaker than the unrestricted form, is a less satisfying model of the dangers inherent in real-life applications of cryptosystems, in that there is no reason to assume that an attacker will be bound by the artificial constraint of only being able to obtain information about various ciphertexts *before* discovering the real target message. Rather, the motivation behind the criterion of security against chosen ciphertext attack is the realistic assumption that attackers may be able to obtain information about chosen ciphertexts through ostensibly honest behaviour, which obviously may occur before or after the sending of a particular target message. On the other hand, some restriction on the definition of the attack is necessary in a non-interactive environment, to rule out playback attacks.

As a less restrictive, reasonably realistic alternative, we propose a model in which each sender possesses a "secret" associated with a publicly-known identifying key. For example, the secret could be the private key associated with the public key by which the sender itself receives messages. In this model, a natural definition of chosen ciphertext attack would allow any party to send chosen ciphertext messages to the receiver; a message's "putative" sender, whose identity is determined by the receiver on the basis of some information in the message, is then provided with its decryption. (In more practical terms, the message's decryption can be assumed to be encrypted using the sender's public key, then broadcast.) Thus the playback attack is not eliminated, but an attacker can still be prevented from getting any useful information from it, except by somehow using it to generate a message of which the attacker itself is the "putative" sender.

The problem of securing messages against chosen ciphertext attack (as defined above) in this model can be solved in a manner similar to that used by Naor and Yung. The solution involves the sender encrypting a message using its own public key and the receiver's, and providing a non-interactive zero-knowledge proof that the two encryptions encrypt the same plaintext. The proof of security of this scheme parallels the proof for the scheme of Naor and Yung.

More generally, we now can define non-interactive zero-knowledge proof of knowledge of a string with some property (say, a witness of membership of another string in some NP language) in this model as a non-interactive zero-knowledge proof of existence of such a string from which the prover can recover the string itself (note that such a proof is with respect to a *specific* prover). For example, an encryption using the prover's public key of a string with the desired property, together with a non-interactive zero-knowledge proof that the encrypted string has that property, would in this model be a non-interactive zero-knowledge proof of knowledge of a string with the property in question. This definition is analogous to (and considerably simpler than) the interactive definition given by Fiat, Feige and Shamir ([FFS]), which required that a string with the property in question be recoverable given "complete access" to the prover TM. It also allows us to restate the above solution to the chosen ciphertext attack problem as a non-interactive version of the interactive protocol of Galil, Haber and Yung. That is, if senders are required to include with each message a non-interactive zero-knowledge proof of the sender's knowledge of its decryption, then the resulting (non-interactive) cryptosystem is secure against chosen ciphertext attack.

A still stronger form of chosen ciphertext attack would allow any message other than the one being attacked to be the chosen ciphertext; its (cleartext) decryption would be returned to the *actual* sender/attacker. Adding the requirement of a digital signature of all complete message-proof pairs guarantees security even in this case, since the sender must prove that anyone holding the information enabling it to sign the messages it generates must also know their decryption. Dolev, Dwork and Naor ([DDN]) have independently presented a considerably more complicated cryptosystem which is also secure against this type of chosen ciphertext attack, and does not make use of senders' possession of a "secret". Any further strengthening of this attack to permit playback attacks would necessarily require some kind of "timeliness" constraint on messages, to prevent their re-use.

2 The Setting and the Problem

We define here a model for a real-life public-key cryptographic setting similar to that first proposed in [DH], in which every party u is provided by a trusted centre with a secret-key public-key pair, (s^u, p^u), such that messages (which we may without loss of generality assume to be single bits) may be encrypted (using a cryptosystem, as described below) with the receiver's public key such that the receiver's private key is needed to decrypt them.

It is assumed that parties are capable of recognizing "valid" public keys (that is, those actually assigned to someone by the trusted centre). It can be imagined that each party possesses a listing of all assigned public keys; alternatively, the centre can provide parties with its own "digital signature" (described later) for its keys (which no party can then forge in attempting to produce a new valid key). A consequence of this state of affairs is that we can now associate with each message a specific "putative" sender (identified by a public key) as well as a receiver. Moreover, the decryption (indeed, the validity or invalidity) of an encrypted message can then be defined so as to depend on the message's putative sender.

In this setting, a chosen ciphertext attack is one in which an attacker, upon observing an encrypted message from another sender to a receiver, is permitted to generate any feasible number of messages, associated with any assigned public key it chooses, and receive their correct decryptions, according to the receiver. If it can with any significant probability greater than $\frac{1}{2}$ decrypt the intercepted message after this process, it has succeeded in its attack.

We can define the attack in one of two ways, depending on how we view the power of parties to the system to learn from other parties' actions and reactions. We can assume that parties are generally only privy to the consequences of the messages that clearly come from themselves; we would in that case correspondingly provide the chosen ciphertext attacker with only the decryptions of messages for wnich the putative sender is the attacker. Alternatively, we can assume that parties are able to detect the reactions of other parties to any messages sent, and to influence (or attempt to simulate) messages from others. The corresponding chosen ciphertext attack would then allow the attacker to receive the correct decryption of any message the attacker generates, regardless of the putative sender (and, in particular, of whether or not the attacker possesses the secret key of the putative sender). We must in this case of course make an exception in the case of the exact message originally being "attacked", but no other exception is necessary.

Note. All polynomial-time TM's defined in this paper are assumed to receive, in addition to their stated inputs, a security parameter 1^n.

Definition 2.1 *A* cryptosystem *is a triple* (M_C, M_E, M_D) *of polynomial time-bounded TM's* (M_C *and* M_E *probabilistic), such that on input u (the "identity" of user u)* M_C *outputs a* secret-key public-key pair (s^u, p^u) *with the following property: On input* (s^u, p^u, p^v, b), $b \in \{0,1\}$, M_E *outputs a string e such that on input* (s^v, p^v, p^u, e) M_D *outputs b. This holds for any two pairs* (s^u, p^u) *and* (s^v, p^v).

Note that u and v might not be distinct; parties may wish to send encryptions to themselves (in fact, this feature will be used later). Note also that "Conventional" public key cryptography (e.g., as defined in [GM]) ignores the identity of the sender altogether; the encryption $M_E(s, p, p', b)$ can be computed by any party without knowledge of any s.

We now define a very general type of chosen ciphertext attack, which in no way restricts the attacker's choice of ciphertexts; such an attack is impossible to defend against, because it permits "playback" attacks. The subsequent definitions define the two ways of restricting the attacker's choice of ciphertext appropriate to our model, as described above.

Definition 2.2 *A* general chosen ciphertext attack *on a cryptosystem* (M_C, M_E, M_D) *is an interactive pair* (A, U), *where A is a probabilistic polynomial-time "attacker", and U represents the "universe" which responds to A's challenges.* (A, U) *behaves as follows: initially,* M_C *creates pairs* (s^S, p^S), (s^R, p^R) *and* (s^A, p^A) *for the sender, receiver and attacker respectively.* M_C *also generates a polynomial number of additional pairs* $(s^1, p^1), \ldots, (s^{q(n)}, p^{q(n)})$ *(representing possible confederates for A). The plaintext* b_S *is then randomly chosen from* $\{0,1\}$, *and* M_E *is run on input* (s^S, p^S, p^R, b_S) *to obtain the encryption e. A is given as input* (e, K), $K = \{p^S, p^R, s^A, p^A, s^1, p^1, \ldots, s^{q(n)}, p^{q(n)}\}$. *A then repeatedly generates triples* (p^i, p^j, m), $p^i, p^j \in K$; *for each such triple, U returns* $M_D(s^j, p^j, p^i, m) \in \{0, 1, \text{“?″}\}$. *At the end of this interaction, A outputs* $b \in \{0,1\}$. *We say that* (A, U) succeeds *if for some polynomial* $h(n)$, $b = b_S$ *with probability at least* $\frac{1}{2} + \frac{1}{h(n)}$, *for infinitely many n.*

Definition 2.3 *An* attacker-specific chosen ciphertext attack *is identical to a general chosen ciphertext attack (as defined above), except that on receiving a triple (p^i, p^j, m), U always returns "?" when $i \neq A$.*

Definition 2.4 *A* message-restricted chosen ciphertext attack *is identical to a general chosen ciphertext attack (as defined above), except that on receiving a triple (p^i, p^j, m), U always returns "?" when $i = S$, $j = R$ and $m = e$.*

3 Tools

3.1 Public-Key Cryptography

We review here the definition of security for public-key cryptosystems introduced in [GM]; A proposed implementation, based on the assumption of the existence of a trapdoor function, is described there.

Definition 3.1 *(based on* [GM]*) A* (single-encryption) attack *against a cryptosystem (M_C, M_E, M_D) is a probabilistic polynomial-time machine A which, on receiving as input (p^u, p^v, e) (where (s^u, p^u) and (s^v, p^v) are generated by M_C, and e is generated by M_E on input (s^u, p^u, p^v, b), with b chosen randomly) outputs a single-bit "guess" g. We say that A* succeeds *if for some polynomial $h(n)$, $b = g$ with probability at least $\frac{1}{2} + \frac{1}{h(n)}$ for infinitely many n; if no such A succeeds, then the cryptosystem is* secure *(against single-encryption attack).*

Note again that u and v need not be distinct; $M_E(s^u, p^u, p^u, b)$ should still be difficult to decrypt for pairs (s^u, p^u) generated by M_C.

3.2 Non-interactive proof systems and zero-knowledge

Non-interactive (zero-knowledge) proof systems were introduced by Blum, Feldman and Micali ([BFM]); the definitions were refined by Bellare and Goldwasser ([BG]) and Naor and Yung ([NY]). We briefly and informally discuss the definitions here, and refer the reader to [NY] for a rigorous treatment of them. An implementation method for any language in NP based on the assumption of the existence of trapdoor functions can be found described in [FLS]. Another implementation appears in [DY].

A *non-interactive proof system* for an NP language L is a pair (P, V) of probabilistic Turing machines, such that given a random bit string r of polynomial length, P can with overwhelmingly high probability construct a "proof" string π for any "theorem" $l \in L$ which causes V to accept (r, l, π), and with only negligible probability will there be an $\bar{l} \notin L$ such that for some "false proof" π, V accepts $(r, \bar{l}, \pi)$. Moreover, V runs in polynomial time, and P runs in polynomial time given a witness w that $l \in L$.

Such a non-interactive proof system is *zero-knowledge* if, for every probabilistic polynomial-time "theorem generator" T which observes a random bit string r and interactively generates members l_i of L and observes their proofs, there exists a "simulator" M which, having chosen the fixed "random" bit string r' according to some distribution, interacts with T to generate strings π_i such that the distribution on strings $r', l_1, \pi_1, \ldots$ generated by T and M is indistinguishable (to non-uniform polynomial-time distinguishers) from the distribution on strings $r, l_1, \pi_1, \ldots$ generated by T and P after receiving a truly random bit string r.

Note that a non-interactive zero-knowledge proof system, as described above, allows repeated use of a single set of public random bits. Hence, the zero knowledge property must take into account the possibility that the theorems which are being proved are chosen *after* the public bits have been revealed, and other theorems have already been proven. The simulator must therefore also allow an adversarial "theorem generator" T to decide what proofs must be simulated based on the simulated pseudorandom public bits and previous simulated proofs.

We can also define a non-interactive zero-knowledge proof of knowledge, analogous to the interactive zero-knowledge proof of knowledge introduced in [FFS]. The definition of the non-interactive version turns out to be much simpler than that for the interactive version, although it assumes a more sophisticated setting. Intuitively, the non-interactive zero-knowledge proof of knowledge is a non-interactive zero-knowledge proof of language membership from which, given certain information, a witness of language membership can be recovered. That information will, in our application, be a secret key associated with a sender's public key (as generated by, say, a trusted centre M_C) in a secure public-key cryptosystem.

Definition 3.2 *Let $L \in NP$, and let W be a polynomial-time "witness-recognizing" machine which accepts some polynomial-sized input of the form (w, l) if and only if $l \in L$. Let $\{D_n\}$ be a family of distributions on polynomial-length strings of the form (s, p). A* non-interactive proof-of-knowledge system *for witnesses of membership in L, relative to $\{D_n\}$ is a non-interactive proof system (P, V) for L in which P receives as its random bit input the string (s, p) chosen according to D_n, while V just receives p as its random* *bit input. Moreover, there must exist a polynomial-time "witness-finder" F with the following property: with "overwhelming probability" (that is, with probability $1 - \frac{1}{g(n)}$, where $-\log g(n) \in \omega(\log n)$), (s,p) will be such that for every l, and for every string π, if V accepts (p,l,π) then $F((s,p),l,\pi)$ is a witness of membership of l in L. The definition of zero-knowledge for proof-of-knowledge systems is essentially the same as for proof systems, except that the theorem generator T receives p as its random bit string inputs, and the simulator M is permitted to generate the p given to T.*

3.3 Digital signature

Informally, a digital signature is a string σ which identifies another string D (the *document*) as originating from the possessor of a particular secret-key public-key pair $(\hat{s}^u, \hat{p}^u)$. A signature scheme is defined by a *key generation algorithm* which produces the secret-key public-key pairs, a *signing algorithm* which given $(\hat{s}^u, \hat{p}^u)$ can generate a signature for any document, and a *checking algorithm*, which uses $\hat{p}^u$ to distinguish valid signatures from invalid ones.

The standard security criterion for a digital signature scheme is security against what is known as *adaptive chosen message attack*. In this attack, the attacker, given $\hat{p}^u$, is permitted to choose any feasible number of documents, and obtain valid signatures for them (interactively, ie. receiving one before choosing the next), before attempting to generate a signature for any document of its choice (distinct from the set of documents whose signatures were obtained). The scheme is secure against this attack if any such attacker succeeds in generating a valid signature for this last document with negligible probabillity.

A rigorous definition for signature schemes and chosen message attack can be found in [GMRi], together with an implementation based a complexity-theoretic assumption. Necessary and sufficient conditions for signature schemes can be found in [R].

4 Non-Interactive Zero-Knowledge Proofs of Knowledge and A Cryptosystem Secure against Chosen Ciphertext Attack

We now show how a secure cryptosystem (in the sense of Definition 3.1 above) and a non-interactive zero-knowledge proof system for NP can be used to build a new cryptosystem secure against chosen ciphertext attack. Encryptions in the new cryptosystem consist of encryptions in the old cryptosystem accompanied by non-interactive zero-knowledge proofs of the sender's knowledge of the decryption (as well as a digital signature, in the case of the stronger message-restricted attack). The proof of knowledge simply consists of encryptions, using the sender's public key, of the random bits used in the original encryption, together with a non-interactive zero-knowledge proof that they are in fact just that. The random bits necessary for the non-interactive proof can be assumed provided by the trusted centre that generates public keys (for example, such bits might be appended to each user's public key). These ideas resemble those used by Naor and Yung in [NY] in their setting.

Theorem 4.1 *Assume that there exist trapdoor functions, as needed for the construction of non-interactive zero-knowledge proofs in [FLS] and of secure cryptosystems (in the sense of definition 3.1) in [GM]; let $\kappa = (M_C, M_E, M_D)$ be such a secure cryptosystem. Then for every NP language L (with witness recognizer W) there exists a non-interactive zero-knowledge proof of knowledge for witnesses of membership in L relative to the distribution on strings $(s,(p,r))$ generated by running M_C, producing (s,p), and appending a random bit string r to p.*

Proof (sketch): The prover uses M_C to generate a new secret-key public key pair (s',p'), encrypts the witness by running M_E on input (s',p',p,w_i) (for each bit w_i of witness w) to produce e, and gives a non-interactive zero-knowledge proof that e can be so constructed. (Actually, for technical reasons, the prover should prove that either e can be so constructed, or a particular otherwised unused portion of r is in fact producible by a particular pseudorandom number generator G (as in [FLS]). The simulator M' of the proof-of-knowledge system generates the extra pseudorandom bits using G, along with an encryption of an arbitrary string, and produces a legitimate non-interactive zero-knowledge proof of the above (true) theorem, using as a witness the seed input into G. Thus the theorems whose proofs are sought from M are always true, and the simulation is therefore indistinguishable from authentic ones.) Full details will be provided in the final paper. ∎

Theorem 4.2 *If there exist trapdoor functions, as needed for the construction of non-interactive zero-knowledge proofs in [FLS] and of secure cryptosystems (in the sense of definition 3.1) in [GM], then there exists a cryptosystem secure against attacker-specific chosen ciphertext attack.*

Proof (sketch): Given the secure cryptosystem $\kappa = (M_C, M_E, M_D)$, consider the new cryptosystem (N_C, N_E, N_D) defined as follows:

1. N_C is identical to M_C, except that it appends a random bit string r^u, of appropriate size for zero-knowledge proofs, to the end of each public key p^u;

2. N_E, given input $(s^u, (p^u, r^u), (p^v, r^v), b)$, runs M_E on input (s^u, p^u, p^v, b) to produce e, and appends to it a "proof" π generated by P in (P, V), a non-interactive zero-knowledge proof of knowledge (with $(s^u, (p^u, r^u))$ as input to P) of a witness that e was so encrypted. N_E outputs (e, π).

3. On input $(s^v, p^v, p^u, (e, \pi))$, N_D runs M_D, obtaining b, and outputs b if V of the proof system above (given (p^u, r^u) as random bit input) accepts input π, and "?" otherwise.

Then no attacker-specific chosen ciphertext attack succeeds against (N_C, N_E, N_D). From any successful such attack, a "breaking" algorithm B can be constructed, using methods similar to those used in [NY], which proves the insecurity of (M_C, M_E, M_D), as follows: given cryptosystem κ, and (p^S, p^R, e) (the sender's public key, receiver's public key, and the encryption being attacked), B generates (s^A, p^A) (the "attacker's" keys), as well as "confederates' keys" (s^1, p^1), and so on, and produces π, a simulated non-interactive zero-knowedge proof of knowledge, relative to the distribution on (s^S, p^S), of a witness that e is a valid encryption under κ (ie., is output by M_E). The algorithm B then simulates *both* A and U ("attacker" and "universe")in the chosen ciphertext attack, taking advantage of the fact that challenges from A containing valid zero-knowledge proofs of knowledge can easily be decrypted by B using s^A and witness-finder F (note also that if B could somehow manage to produce valid-looking encryptions with putative sender A, containing convincing but invalid zero-knowledge proofs of knowledge—a feat which would certainly be impossible, except with negligible probability, if π were not simulated—then there would exist a means of distinguishing simulated proofs-of-knowledge from real ones). Full details of the proof will appear in the final paper. ▌

Theorem 4.3 *If there exist trapdoor functions, as needed for the construction of non-interactive zero-knowledge proofs in [FLS] and of secure cryptosystems (in the sense of definition 3.1) in [GM], then there exists a cryptosystem secure against message-restricted chosen ciphertext attack.*

Proof (sketch): Consider a cryptosystem (N_C, N_E, N_D) defined as in Theorem 4.2 with the following modifications:

1. N_C runs M_C and the key generation algorithm for a secure signature scheme on input u to produce a secret-key public-key pair $((s^u, \hat{s}^u), (p^u, \hat{p}^u, r^u))$.

2. On input $((s^u, \hat{s}^u), (p^u, \hat{p}^u, r^u), (p^v, \hat{p}^v, r^v), b)$, N_E appends to its output (e, π) (generated using M_E, as described in theorem 4.2) a digital signature for it using the private signature key $\hat{s}^u$.

3. On input $((s^v, \hat{s}^v), (p^v, \hat{p}^v, r^v), (p^u, \hat{p}^u, r^v), (e, \pi, s))$, N_D obtains b using M_D as described in theorem 4.2, and outputs b if V (as described in Theorem 4.1) accepts input π and the checking algorithm of the digital signature scheme accepts the appended digital signature, and "?" otherwise.

Then no message-specific chosen ciphertext attack succeeds against (N_C, N_E, N_D). The proof is similar to that of theorem 4.2, except that the security of signature schemes against chosen message attack is used to show that the attacker has negligible probability of generating a valid message which is not just as easily decryptable by the simulated U as those permissible under an attacker-specific attack. Full details will be presented in the final paper. ▮

Note that the cryptosystems described in theorems 4.2 and 4.3 can both easily be proven, in a similar manner, to be secure against chosen plaintext attack, as well. The proof is necessary, however, because the sender's keys are involved in the encryption process in our revised model (in more "conventional" public-key cryptography, in contrast, chosen-plaintext security follows automatically from the security property described in definition 3.1). The new model therefore raises the issue, for a time somewhat obscured by the prominence of public-key cryptography, of just what kinds of attack a cryptosystem should resist; if the chosen-plaintext and chosen-ciphertext attacks are in some sense incomparable, and must be dealt with separately, then there may be other natural attacks, as well, security against which is guaranteed by neither chosen-plaintext nor chosen-ciphertext security.

5 Conclusions

The choice of an appropriate definition and solution to the problem of chosen ciphertext attack can be considered as a first step toward formalizing and solving the much broader problem of securing public-key cryptographic communication in a multiparty setting. Many other issues need to be resolved, including the development of a rigorous definition of security, consideration of the problem of traffic analysis, and examination of the apparently necessary task of implementing timeliness constraints on messages.

References

[BFM] M. Blum, P. Feldman, and S. Micali, *Non-Interactive Zero Knowledge and its Applications*, Proc. 20th ACM Symposium on Theory of Computing (1988), pp. 103–112.

[BG] M. Bellare and S. Goldwasser, *New Paradigms for Digital signatures and Message Authentication based on Non-Interactive Zero-Knowledge Proofs*, Proc. CRYPTO '89.

[DDN] D. Dolev, C. Dwork and M. Naor, *Non-Malleable Cryptography*, Proc. 23rd ACM Symposium on Theory of Computing (1991), pp. 542–552.

[DH] W. Diffie and M. Hellman, *New directions in Cryptography*, IEEE Trans. on Information Theory 22(6), 1976, pp. 644–654.

[DY] A. De Santis and M. Yung, *Cryptographic Applications of the Non-Interactive Metaproof and Many-Prover Systems*, Proc. CRYPTO '90.

[FFS] U. Feige, A. Fiat, and A. Shamir, *Zero Knowledge Proofs of Identity*, Proc. 19th ACM Symp. on Theory of Computing (1987), pp. 210–217.

[FLS] U. Feige, D. Lapidot and A. Shamir, *Multiple Non-Interactive Zero-Knowledge Proofs Based on a Single Random String*, Proc. 31st IEEE Symp. on Foundations of Computer Science (1990), pp. 308–317.

[GHY] Z. Galil, S. Haber and M. Yung, *Symmetric Public-Key Cryptosystems*, submitted to J. of Cryptology.

[GM] S. Goldwasser and S. Micali, *Probabilistic Encryption*, JCSS Vol. 28, No. 2 (April 1984), pp. 270–299.

[GMRa] S. Goldwasser, S. Micali and C. Rackoff, *The Knowledge Complexity of Interactive Proof Systems*, Proc. 17th ACM Symp. on Theory of Computing (1985), pp. 291–304.

[GMRi] S. Goldwasser, S. Micali and R. Rivest, *A Secure Digital Signature Scheme*, SIAM J. on Computing, Vol. 17, 2 (1988), pp. 281–308.

[NY] M. Naor and M. Yung, *Public-Key Cryptosystems Provably Secure Against Chosen Ciphertext Attacks*, Proc. 22nd ACM Symp on Theory of Computing (1990), pp. 427–437.

[R] J. Rompel, *One-Way Functions Are Necessary and Sufficient for Secure Signatures*, Proc. 31st IEEE Symp. on Foundations of Computer Science (1990), pp. 387–394.