# Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret (Extended Abstract)

Josh Cohen Benaloh*

**Abstract**

In 1979, Blackley and Shamir independently proposed schemes by which a secret can be divided into many shares which can be distributed to mutually suspicious agents. This paper describes a homomorphism property attained by these and several other secret sharing schemes which allows multiple secrets to be combined by direct computation on shares. This property reduces the need for trust among agents and allows secret sharing to be applied to many new problems. One application described here gives a method of verifiable secret sharing which is much simpler and more efficient than previous schemes. A second application is described which gives a fault-tolerant method of holding verifiable secret-ballot elections.

## 1 Introduction

Suppose that Alice holds a secret $A$ and distributes shares of her secret to $n$ agents, using Shamir's secret sharing (threshold) scheme ([Sha79]), such that any $k$ agents can construct $A$. Suppose further that Bob holds a secret $B$ and distributes shares of $B$ to the same $n$ agents in the same way as Alice. Finally suppose that $k$ of the agents decide that they want to determine $A + B$ while revealing as little information about $A$ and $B$ as possible. (Of course, revealing $A + B$ yields some partial information about $A$ and $B$.) How can this be done?

It is not hard to see that if each of the $k$ agents releases the sum of the two shares it holds, each of these sums is itself a share of the sum of the secrets $A + B$. In short, the sum of the shares of the secrets are shares of the sum of the secrets. It is also the case that release of these share sums gives no information about $A$ and $B$ other than that contained in the release of their sum $A + B$.

In general, suppose each of $m$ parties holds a "sub-secret", and there exists a "super-secret" which is the composition of the sub-secrets under some known function (such as the sum or the product of the sub-secrets). The parties want

*This work was supported in part by the National Security Agency under Grant MDA904-84-H-0004.

to determine the super-secret without revealing their sub-secrets and without depending upon cryptographic assumptions.

Cryptographic techniques for computing with encrypted data have been studied in [RAD78], [DLM82], [Yao82], [BlMe85], and [Fei85], for example. This approach to the problem, however, depends heavily upon cryptographic assumptions such as the difficulty of factoring. In this paper, we shall consider an alternate approach to such problems in which no cryptography or cryptographic assumptions are required (although the data used may be encrypted for other reasons).

With an appropriate secret sharing homomorphism, shares of the sub-secrets can be distributed to $n$ agents such that any $k$ can determine each of the sub-secrets. Each agent can then compose its "sub-shares" into a single "super-share" such that any $k$ of the super-shares are sufficient to determine the super-secret.

The advantage of such a homomorphism is that $k$ of the $n$ agents can, by revealing their super-shares, determine the super-secret without sharing any information about the constituent sub-secrets. Information about the sub-secrets can only be obtained if $k$ or more agents agree to merge their sub-shares to reconstruct the sub-secrets.

At this point, we assume that there are no attempts at subversion. The information is assumed to be correct, and the only concern is that some of the agents may surreptitiously collaborate in order to obtain secret information. In section 4, we shall see examples of how interactive proofs and cryptographic methods can be used to verify both the validity of the shares given to the agents and the accuracy of the composite results returned by the agents.

Two applications of this homomorphism will be seen.

The first allows the validity of secret shares to be verified without their being revealed. Here, a shareholding agent can obtain very high confidence that it holds a valid share of the secret rather than a useless random number. A share is valid if it, when combined with *any* other $k-1$ shares, yields the same secret as does any subset of $k$ of the shares.

The second application is in the domain of elections. Here a voter can distribute shares of his or her vote to $n$ agents. Each agent can then compose its vote-shares to form a share of the election tally. If $k$ or more of the agents reveal their composite tally-shares, then the election tally is publically revealed. A conspiracy of at least $k$ dishonest agents is required, however, in order to obtain information about an individual vote.

# 2 The Homomorphism Property

Shamir in [Sha79] defines a $(k, n)$ *threshold scheme* to be a division of a secret $D$ into $n$ pieces $D_1, \ldots, D_n$ in such a way that:

**(1)** knowledge of any $k$ or more $D_i$ pieces makes $D$ easily computable;

**(2)** knowledge of any $k-1$ or fewer $D_i$ pieces leaves $D$ completely undetermined (in the sense that all its possible values are equally likely).

Let $S$ be the domain of possible secrets, and let $T$ be the domain of legal shares. Every instance of a $(k,n)$ threshold scheme determines a set of functions $F_I : T^k \rightarrow S$ defined for each $I \subseteq \{1,2,\ldots,n\}$ with $|I| = k$. These functions define the value of the secret $D$ given any set of $k$ values $D_{i_1},\ldots,D_{i_k}$:

$$D = F_I(D_{i_1},\ldots,D_{i_k}),$$

where $I = \{i_1,\ldots,i_k\}$.

**Definition** Let $\oplus$ and $\otimes$ be binary functions on elements of the secret domain $S$ and of the share domain $T$, respectively. We say that a $(k,n)$ threshold scheme has the $(\oplus,\otimes)$-*homomorphism* property (or is $(\oplus,\otimes)$-*homomorphic*) if for all $I$, whenever

$$D = F_I(D_{i_1},\ldots,D_{i_k})$$

and

$$D' = F_I(D'_{i_1},\ldots,D'_{i_k}),$$

then

$$D \oplus D' = F_I(D_{i_1} \otimes D'_{i_1},\ldots,D_{i_k} \otimes D'_{i_k}).$$

This property implies that the composition of the shares are shares of the composition.

It is easy to see that Shamir's polynomial based secret sharing scheme is $(+,+)$-homomorphic, but it is not quite so apparent that Shamir's scheme satisfies another property which is also necessary to capture the intuition described earlier. We want it to also be the case that up to $k-1$ sets of sub-shares together with *all* of the super-shares (and therefore the super-secret) give no more information about the sub-secrets than does the super-secret alone.

Shamir's definition of a $(k,n)$ threshold scheme does not allow for such partial information, but Kothari in [Kot84] generalizes Shamir's definition slightly to allow for the possibility of *a priori* information about a secret. Kothari's definition can be summerized by replacing condition (2) above with

**(2′)** $\text{Prob}(D = x) = \text{Prob}(D = x \mid D_{i_1} = x_{i_1}, D_{i_2} = x_{i_2},\ldots,D_{i_{k-1}} = x_{i_{k-1}})$ for an arbitrary set of $k-1$ indices $\{i_1,i_2,\ldots,i_{k-1}\}$ for all $x \in S$ and all $x_i \in T$.

This says that even with partial *a priori* information about the secret $D$, possession of up to $k-1$ shares of $D$ gives no additional information about the value of $D$.

It is now possible to give a formal definition to capture the intuition that no extraneous information is released by a secret sharing homomorphism.

**Definition** We define a $(\oplus, \otimes)$-*composite* $(k, n)$ *threshold scheme* to be a division of a set of $m$ *sub-secrets* $d_1, \ldots, d_m$ into *sub-shares* $d_{i,j}, 1 \leq i \leq n, 1 \leq j \leq m$ ($d_{i,j}$ is the $i^{\text{th}}$ share of the $j^{\text{th}}$ sub-secret $d_j$) such that

**(1)** The *super-secret* $D = d_1 \oplus d_2 \oplus \cdots \oplus d_m$ is easily computable given $k$ or more distinct *super-shares* $D_i = d_{i,1} \otimes d_{i,2} \otimes \cdots \otimes d_{i,m}$;

**(2)** For all possible values $X \in S$ of the super-secret $D$ and for every possible value $x_j$ of each sub-secret $d_j$ $(1 \leq j \leq M)$,

$$\text{Prob}(d_j = x_j \mid D = X) =$$
$$\text{Prob}(d_j = x_j \mid D = X; \forall i \in I, D_i = X_i; \forall i \in I', \forall j \in J, d_{i,j} = x_{i,j})$$

where $I = \{1, 2, \ldots, n\}$, $J = \{1, 2, \ldots, m\}$, and $I'$ is an arbitrary subset of size up to $k - 1$ of $\{1, 2, \ldots, n\}$ and for all possible values $X_i \in T$ of the super-shares and for every possible value $x_{i,j} \in T$ of the sub-shares.

Intuitively, the first property says that any $k$ of the $n$ agents can together determine the super-secret $D$. The second property asserts that no conspiracy of fewer than $k$ agents can gain any information at all about any of the sub-secrets $d_j$ (other than that already given by the super-secret $D$) even when given all of the super-shares $D_i$.

The following theorem is somewhat surprising,

**Theorem 1** *If the secret domain $S$ and the share domain $T$ are finite and of the same cardinality, then every $(\oplus, \otimes)$-homomorphic $(k, n)$ threshold scheme is a $(\oplus, \otimes)$-composite $(k, n)$ threshold scheme.*

*Proof:* (sketch)

The definition of $(\oplus, \otimes)$-homomorphism implies condition (1) immediately.

To prove condition (2), it is simpler to consider only the case when $m = 2$ (two sub-secrets). The case for arbitrary $m$ follows straightforwardly.

Consider a table of the form

| $S$ | $s_1$ | $s_2$ | $\cdots$ | $s_n$ |
|---|---|---|---|---|
| $A$ | $a_1$ | $a_2$ | $\cdots$ | $a_n$ |
| $B$ | $b_1$ | $b_2$ | $\cdots$ | $b_n$ |

where $S = A \oplus B$ and for all $i$, $s_i = a_i \otimes b_1$. $S$ is the secret defined by the shares $s_1, \ldots, s_n$, $A$ is the secret defined by the shares $a_1, \ldots, a_n$, and $B$ is the secret defined by the shares $b_1, \ldots, b_n$,

Assume that a set of up to $k - 1$ conspirators are willing to share some of their information in order to try to gain information about $A$ and $B$. Without loss of generality, assume that these conspirators are among the first $k - 1$ shareholders.

By the definition of a $(k, n)$ threshold scheme, $A$ and $B$ remain completely undetermined even if $k - 1$ shares are known. Therefore, we may assume that

all of $a_1, a_2, \ldots, a_{k-1}$ and $b_1, b_2, \ldots, b_{k-1}$ are known to the conspirators. With this information, the conspirators are able to compute $s_1, s_2, \ldots, s_{k-1}$ without assistance. It is already assumed that the "super-secret" $S$ is known. Therefore, since $|S| = |T| < \infty$, the "super-shares" $s_k, s_{k+1}, \ldots, s_n$ are completely determined and can be computed by the conspirators. Thus, their release to the conspirators gives them no additional information. ∎

**Remark** The condition that the secret domain $S$ and the share domain $T$ are of the same finite cardinality was not strictly required, and the following weaker property will suffice. For a given super-share $D_k$ and sub-secrets $d_1$,$d_2$,$d'_1$,$d'_2$, such that $d_1 \oplus d_2 = D = d'_1 \oplus d'_2$, let $p$ be the conditional probability that $D_k = d_{1,k} \otimes d_{2,k}$ for some $d_{1,k}$ and $d_{2,k}$ which imply sub-secrets $d_1$ and $d_2$, respectively, and let $p'$ be the conditional probability that $D_k = d'_{1,k} \otimes d'_{2,k}$ for some $d'_{1,k}$ and $d'_{2,k}$ which imply sub-secrets $d'_1$ and $d'_2$, respectively. If $p = p'$ for all such $d_1$, $d_2$, $d'_1$, $d'_2$, and $D_k$, then the conclusion of the theorem is true.

For simplicity of exposition (and to keep the notation under control), this generalization has not been incorporated into the theorem. Its inclusion is straightfoward, but cumbersome, and appears to offer no additional insights.

# 3 Some Examples

It is easy to see that the properties of polynomials give Shamir's $(k, n)$ threshold scheme the $(+, +)$-homomorphism property, and since the secret domain and the share domain consist of the same finite set (namely the integers modulo $p$), Shamir's scheme is a $(+, +)$-composite $(k, n)$ threshold scheme and enjoys all of the properties thereof.

Some other techniques can also be easily seen to produce $(+, +)$-composite $(k, n)$ threshold schemes. See [Bla79], [AsBl80], and [Kot84] for some further examples.

What if the super-secret is not the sum of the sub-secrets? Shamir's scheme is *not* $(\times, \times)$-composite. This is because the product of two non-constant polynomials is of higher degree than the factors.

By using a homomorphism between addition and discrete logarithms, for example, it *is* possible to transform Shamir's scheme into a $(\times, +)$-composite $(k, n)$ threshold scheme. Thus, if the desired super-secret is the product of the sub-secrets, Shamir's scheme can still be used. This method can be summarized by the following adage. *The sum of the shares of the discrete logs of the secrets are shares of the discrete log of the product of the secrets.*

In general, discrete logarithms may be difficult to compute. However, if $p$ is small or of one of a variety of special forms, the problem is tractable (see [PoHe78], [Adl79], [COS86]). It should be emphasized that such special cases for $p$ do not in any way weaken the security of our schemes. The security is *not* cryptographic,

but rather is information theoretic. Therefore, there need be no assumptions about the difficulty of solving any special problems.

# 4 Applications

The applications described here rely on the encryption of shares both to fascilitate their distribution and allow for a mechanism which ensures that certrain properties of the shares are attained. Since the encryptions of shares are made public, the security is no longer information theoretic, but rather depends upon a cryptographic assumption.

The encryption function used here was introduced in [CoFi85] and draws upon the ideas of probabilistic encryption found in [GoMi84]. The function is also described in [BeYu86].

Before beginning, a prime number $r$ is fixed such that $r \geq |S|$ — the size of the secret domain. To develop an encryption function $E$, one selects primes $p$ and $q$ such that $r|(p-1)$ and $r\not|(q-1)$. Let $N$ be the product $N = pq$. The developer releases the pair $(N, y)$ where $y$ is relatively prime to $N$ and $y$ is *not* an $r^{\text{th}}$ residue modulo $N$.[1] It is necessary in most applications for the developer of such an encryption function to convince others that $y$ is, in fact, not an $r^{\text{th}}$ residue modulo $N$. This may be accomplished by interactive proof techniques described in [CoFi85] and [BeYu86].

To use $E$ to encrypt a value $s$, one randomly selects an $x$ and forms $E(s, x, y, N) = y^s x^r \bmod N$. The holder of the trapdoor factors of $N$ can easily determine $s$ from $E(s, x, y, N)$. However, there is no known efficient method for determining $s$ from its encryption when the factors of $N$ are not known.

## 4.1 Verifiable Secret Sharing

The first application gives a simple and efficient method for verifiable secret sharing. This problem was first described in [CGMA85] and the application of secret sharing homomorphisms to this problem was developed as a result of an observation made by Oded Goldreich.

**Definition** We say that a set of $n$ shares $s_1, s_2, \ldots, s_n$ is *$k$-consistent* if every subset of $k$ of the $n$ shares defines the same secret.

The problem of verifiable secret sharing is to convince shareholders that their shares (collectively) are $k$-consistent.

It is easy to see that in Shamir's scheme, the shares $s_1, s_2, \ldots, s_n$ are $k$-consistent if and only if the interpolation of the points $(1, s_1), (2, s_2), \ldots, (n, s_n)$ yields a poly-

[1] $y$ is an $r^{\text{th}}$ residue modulo $N$ if and only if there exists an $x$ such that $y \equiv x^r \pmod{N}$.

nomial of degree *at most* $d = k - 1$. It is also useful to observe that if the sum of two polynomials is of degree at most $d$, then either both are of degree at most $d$ or both are of degree greater than $d$.

This suggests the following outline of an interactive proof that a polynomial $P$, given by its (encrypted) values at $n$ distinct points, is of degree at most $d$ (see [FMR84] and [GMR85] for a description of interactive proofs and applications).

1. Encryptions of the values of the points that describe $P$ are released by the prover.
2. Encryptions of many (say 100) additional random polynomials again of degree at most $d$ are also released by the prover.
3. A random subset of the random polynomials is designated by the verifier(s).
4. The polynomials in the chosen subset are decrypted by the prover. They must all be of degree at most $d$.
5. Each remaining random polynomial is added to $P$. (Note that pointwise addition gives the same polynomial as the coefficientwise addition.) Each of these sum polynomials is decrypted by the prover. They must also all be of degree at most $d$.

The encryption of the values of each point must be probabilistic (to prevent guessing of values) and satisfy a homomorphism property (so that an encryption of the sum of two values can be developed directly from the encryptions of the two values). These properties are satisfied by the encryption function $E$ described above.

In more detail, a secret $s$ is divided into $n$ shares $s_1, s_2, \ldots, s_n$ such that the polynomial $P$ interpolated through the points $(i, s_i)$ has degree at most $k - 1$ and passes through the point $(0, s)$. (So far, this is precisely Shamir's scheme). Each (future) shareholder selects and makes public an $(N_i, y_i)$ pair to develop an encryption function $E_i$ as above. The $i^{\text{th}}$ share, $s_i$, is transmitted to the $i^{\text{th}}$ shareholder by selecting a random $x_i$ and releasing $E_i(s_i, x_i, y_i, N_i) = y_i^{s_i} x_i^r \bmod N_i$.

To prove interactively that the (encrypted) points released describe a polynomial with degree no more than $d$, prepare (say) 100 more random polynomials, each of degree at most $d$, in exactly the same way. The values of these random polynomials at 0 (the secrets they describe) are also selected randomly.

The verifiers randomly select a subset $A$ of these random polynomials. Each polynomial in $A$ is opened by revealing the corresponding $s_i$ and $x_i$. For each polynomial $P'$ not in $A$, the (pointwise) sum $P + P'$ is opened by releasing $s_i + s_i' \bmod r$ and $x_i \cdot x_i' \cdot y^{\lfloor (s_i + s_i')/r \rfloor}$ where the $i^{\text{th}}$ point of $P'$ is given by $E_i(s_i', x_i', y_i, N_i)$. All points released should describe polynomials of degree at most $d$.

It is not hard to see that a set of random polynomials of degree at most $d$ together with a set of sums of $P$ and other random polynomials of degree at most

$d$ gives no useful information about $P$ (other than that its degree is bounded by $d$).

## 4.2 Secret-Ballot Elections

The motivating application for this work is in the domain of cryptographic elections. In [CoFi85], an election scheme is presented which allows a government to hold an election in which the legitimacy of the votes and the tally is verified by means of interactive proofs.

Although, there is high confidence in the correctness of the tally in such an election, the government is a "trusted authority" with the ability to see every vote and thereby compromise every voter's privacy.

In [BeYu86], the government is replaced by a set of "tellers" such that it is necessary for all tellers to conspire in order to compromise a voter's privacy. In that scheme, however, if even one of the tellers fails to complete its protocol properly, the entire election fails and no tally is produced.

The basic election scheme described in these papers can, however, be embedded within a $(+,+)$-composite $(k,n)$ threshold scheme (in particular, in Shamir's scheme) as suggested by the outline below. This extension is also described in [Coh86].

Instead of a single government, $n$ sub-governments (or *tellers*) each hold a sub-election. Each voter chooses either 0 or 1 as a secret value (0 indicating a **no** vote, 1 indicating a **yes** vote) and distributes one share of the secret vote to each of the $n$ tellers. The tally of the election will be the sum of the voters' secrets.

After votes are cast, each teller simply adds the vote-shares it has received using the (single government) verifiable election scheme of [CoFi85]. Since the $(k,n)$ threshold scheme has the $(+,+)$-homomorphism property, this sum of vote-shares is itself a share of the sum (tally) of the votes. Thus, once $k$ or more tellers release their sub-tallies, the overall election tally can be determined. Furthermore, since the secret domain and the share domain consist of the same finite set, the conditions of Theorem 1 are satisfied, and $k$ or more conspiring tellers are required to determine any individual voter's secret vote.

The interactive proof techniques used in section 4.1 can be generalized slightly to allow verification of the vote-shares. Here, each voter participates in an interactive proof to demonstrate to all participants that the vote-shares it distributes are *legitimate* in the sense that every set of $k$ vote-shares derives the same secret vote and that this vote is either a 0 or a 1.

Thus, as long as at least $k$ of the $n$ designated tellers participate through to conclusion, an election can be conducted such that each participant has very high confidence in the accuracy of the resulting tally and no set of fewer than $k$ tellers (together with any number of conspiring voters) can (without breaking the underlying cryptosystem and thereby solving an open number theoretic problem)

gain more than a polynomially small advantage at distinguishing between possible votes of honest voters.

## Acknowledgements

The author would like to express many thanks to Mike Fischer, Oded Goldreich, Gregory Sullivan, and David Wittenberg for their help in developing this work.

## References

[Adl79] **Adleman, L.** "Subexponential Algorithm for The Discrete Logarithm Problem." *Proc.* 20th *IEEE Symp. on Foundations of Computer Science*, San Juan, PR (Oct. 1979), 55–60.

[AsBl80] **Asmuth, C.** and **Bloom, J.** "A Modular Approach to Key Safeguarding." *Texas A&M University, Departement of Mathematics*, College Station, TX (1980).

[BeYu86] **Benaloh, J.** and **Yung, M.** "Distributing the Power of a Government to Enhance the Privacy of Voters." *Proc.* 5th *ACM Symp. on Principles of Distributed Computing*, Calgary, AB (Aug. 1986).

[Bla79] **Blakley, G.** "Safeguarding Cryptographic Keys." *Proc. AFIPS 1979 National Computer Conference*, New York, NY (June 1979), 313–317.

[BlMe85] **Blakley, G.** and **Meadows, C.** "A Database Encryption Scheme Which Allows the Computation of Statistics Using Encrypted Data." *Proc. IEEE Symposium on Computer Security and Privacy*, Oakland, CA (Apr. 1985).

[CGMA85] **Chor, B., Goldwasser, S., Micali, S.,** and **Awerbuch, B.** "Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults." *Proc.* 26th *IEEE Symp. on Foundations of Computer Science*, Portland, OR (Oct. 1985), 383–395.

[CoFi85] **Cohen, J.** and **Fischer, M.** "A Robust and Verifiable Cryptographically Secure Election Scheme." *Proc.* 26th *IEEE Symp. on Foundations of Computer Science*, Portland, OR (Oct. 1985), 372–382.

[Coh86] **Cohen, J.** "Improving Privacy in Cryptographic Elections." *TR-454, Yale University, Departement of Computer Science*, New Haven, CT (Feb. 1986).

[COS86] **Coppersmith, D., Odlyzko, A.,** and **Schroeppel, R.** "Discrete Logarithms in $GF(p)$." *Algorithmica*, 1 (1986), 1–15.

[DLM82] **DeMillo, R., Lynch, N.,** and **Merritt, M.** "Cryptographic Protocols." *Proc.* 14th *ACM Symp. on Theory of Computing*, San Francisco, CA (May 1982), 383–400.

[Fei85] **Feigenbaum, J.** "Encrypting Problem Instances or Can You Take Advantage of Someone Without Having to Trust Him", *Proc. Crypto '85*, Santa Barbara, CA (Aug. 1985), 477–488. Published as *Advances in Cryptology*, ed. by H. Williams in *Lecture Notes in Computer Science*, vol. 218, ed. by G. Goos and J. Hartmanis. Springer-Verlag, New York (1985).

[FMR84] **Fischer, M., Micali, S.,** and **Rackoff, C.** "A Secure Protocol for the Oblivious Transfer." Presented at *Eurocrypt84*, Paris, France (Apr. 1984). (Not in proceedings.)

[GMR85] **Goldwasser, S., Micali, S.,** and **Rackoff C.** "The Knowledge of Complexity of Interactive Proof-Systems." *Proc.* 17th *ACM Symp. on Theory of Computing*, Providence, RI (May 1985), 291–304.

[GoMi84] **Goldwasser, S.** and **Micali, S.** "Probabilistic Encryption." *J. Comput. System Sci. 28*, (1984), 270–299.

[Kot84] **Kothari, S.** "Generalized Linear Threshold Scheme." *Proc. Crypto '84*, Santa Barbara, CA (Aug. 1984), 231–241. Published as *Advances in Cryptology*, ed. by G. Blakely and D. Chaum in *Lecture Notes in Computer Science*, vol. 196, ed. by G. Goos and J. Hartmanis. Springer-Verlag, New York (1985).

[PoHe78] **Pohlig, S.** and **Hellman, M.** "An Improved Algorithm for Computing Logarithms Over GF(2) and Its Cryptographic Significance." *IEEE Trans. on Information Theory 24*, 1 (Jan. 1978), 106–110.

[RAD78] **Rivest, R., Adleman, L.,** and **Dertouzos, M.** "On Data Banks and Privacy Homomorphisms." *Foundations of Secure Computation*, ed. by R. A. DeMillo, et. al. Academic Press, New York (1978), 169–179.

[Sha79] **Shamir, A.** "How to Share a Secret." *Comm. ACM 22*, 11 (Nov. 1979), 612–613.

[Yao82] **Yao, A.** "Protocols for Secure Computations." *Proc.* 23rd *IEEE Symp. on Foundations of Computer Science*, Chicago, IL (Nov. 1982), 160–164.