J. Cryptology (1988) 1: 133–138

Journal of Cryptology
© 1988 International Association for
Cryptologic Research

How To Share a Secret with Cheaters[1]

Martin Tompa
IBM Thomas J. Watson Research Center, P.O. Box 218,
Yorktown Heights, NY 10598, U.S.A.

Heather Woll
Department of Computer Science, FR35, University of Washington,
Seattle, WA 98195, U.S.A.

Abstract. This paper demonstrates that Shamir's scheme [10] is not secure against certain forms of cheating. A small modification to his scheme retains the security and efficiency of the original, is secure against these forms of cheating, and preserves the property that its security does not depend on any unproven assumptions such as the intractability of computing number-theoretic functions.

Key words. Secret sharing, Cheating, Security, Interpolation, Reconstruction.

1. How To Cheat When Sharing a Secret

Shamir [10] proposed and solved a problem in which a secret known only to one party is to be divided among n other participants. This is to be done in such a way that a certain number k of these participants is necessary and sufficient to reconstruct the secret. Each individual participant knows n, k, and the set of possible values of the secret. The problem is stated more precisely as follows:

Inputs.

- Nonnegative integers n, s, and $k \leq n$.
- A "secret" $D \in \{0, 1, \ldots, s-1\}$.

Problem. Divide D into "shares" $D_1, D_2, \ldots, D_n$ such that:

(a) Knowledge of any k shares is sufficient to reconstruct D efficiently.
(b) Knowledge of any $k-1$ shares provides no more information about the value of D than was known before.

Such a scheme would be useful, for example, when some data must be replicated over n locations (say, for convenience or fault tolerance), and simultaneously must

[1] This material is based in part upon work supported by the National Science Foundation under Grant Nos. DCR-8301212 and DCR-8352093. Part of the work was performed while the second author was a visitor at the IBM Thomas J. Watson Research Center.

be protected from $k-1$ security violations (for example, due to sensitivity of the data or mistrust among the participants).

Shamir's solution is simple, elegant, and, unlike most other protocols related to cryptography, not dependent on any unproven assumptions about the complexity of computing certain number-theoretic functions. Shamir's scheme for dividing D into shares is as follows:

1. Choose any prime $p \geq \max(s, n+1)$. Let Z_p represent the field of integers modulo p.
2. Choose $a_1, a_2, \ldots, a_{k-1} \in Z_p$ randomly, uniformly, and independently.
3. Let $q(x) = D + a_1 x + a_2 x^2 + \cdots + a_{k-1}x^{k-1}$.
4. Let $D_i = q(i)$, for all $1 \leq i \leq n$. (The evaluation of $q(i)$ is done over Z_p.)

Properties (a) and (b) now follow from the interpolation theorem, which states that k points are necessary and sufficient to determine $q(x)$. (Details are given in the next section.) For later reference, the interpolation theorem is stated here. The reader is referred to Lipson [8] for a thorough treatment.

Interpolation Theorem. *For any field F, for any k distinct elements $x_1, x_2, \ldots, x_k \in F$, and for any k elements $y_1, y_2, \ldots, y_k \in F$, there exists a unique polynomial $q(x) \in F[x]$ with degree less than k such that $q(x_i) = y_i$ for all $1 \leq i \leq k$.*

Since the scheme is intended to be useful in applications involving mistrustful participants, the following property is desirable in addition to (a) and (b):

(c) There is only a small probability $\varepsilon > 0$ that any $k-1$ participants $i_1, i_2, \ldots, i_{k-1}$ can fabricate new shares $D'_{i_1}, D'_{i_2}, \ldots, D'_{i_{k-1}}$ that deceive a kth participant i_k. Here, deceiving the kth participant means that, from $D'_{i_1}, D'_{i_2}, \ldots, D'_{i_{k-1}}$, and D_{i_k}, the secret D' reconstructed is "legal" (i.e., $D' \in \{0, 1, \ldots, s-1\}$), but "incorrect" (i.e., $D' \neq D$).

The desirability of condition (c) is particularly clear when $k = 2$. Without condition (c), a cheater can obtain D while simultaneously, and without being detected, convincing a second participant of an incorrect secret.

Notice the stronger version of condition (c) resulting when $\varepsilon = 0$ is unattainable. This is due to the fact that condition (b) implies that, for any share D_{i_k} of the secret D and any legal but incorrect secret $D' \neq D$, there must exist $D'_{i_1}, D'_{i_2}, \ldots, D'_{i_{k-1}}$ such that the collection of shares $\{D'_{i_1}, D'_{i_2}, \ldots, D'_{i_{k-1}}, D_{i_k}\}$ represents the secret D', thus deceiving the kth participant.

Unfortunately, Shamir's scheme is not secure against such cheating. Firstly, if $p = s$ then all reconstructed secrets are legal, so that it is impossible for the kth participant to detect cheating. We might guess from this that Shamir's scheme can be made secure by choosing p much greater than s, since then there would be only a slight chance of the reconstructed secret being legal. The following example shows that this is not the case. In fact, with high probability a *single* participant can deceive $k-1$ others.

Suppose that participants $i_1, i_2, \ldots, i_k$ agree to pool their shares. Participant i_1, who decides to cheat, uses interpolation to find a polynomial $\Delta(x)$ of degree at most $k-1$ such that $\Delta(0) = -1$ and $\Delta(i_2) = \Delta(i_3) = \cdots = \Delta(i_k) = 0$. Having been given the share D_{i_1}, participant i_1 announces instead the share $D_{i_1} + \Delta(i_1)$. Now the interpolation theorem guarantees that the k participants will reconstruct the polynomial $q(x) + \Delta(x)$, which has constant term $q(0) + \Delta(0) = D - 1$. Thus, the deception will go undetected unless the original secret happened to be $D = 0$.

In the next section it is shown that a small modification of Shamir's scheme has all three properties (a), (b), and (c). (In fact, even knowledge of both the secret D and the polynomial $q(x)$ does not increase the probability of successful deception.) Furthermore, the expected running time is polynomial in k, n, $\log s$, and $\log(1/\varepsilon)$.

One straightforward solution to the problem of cheating is to have the distributor of shares sign each share D_i with an unforgeable signature (such as that proposed in [7]). This is, in fact, exactly the solution that Rabin [9] chose when he used Shamir's scheme to solve the problem of agreement among distributed processes that might cheat. There are two advantages of our scheme over the use of Shamir's scheme plus signatures:

1. All currently known signature schemes depend upon such unproven hypotheses as the intractability of integer factorization, whereas our secret sharing scheme, like Shamir's, does not. In fact, our scheme is secure even if the conspirators have unlimited computational resources.
2. Our scheme is exactly as easy to implement as Shamir's, thus avoiding the complications of implementing an additional signature scheme.

A recent paper [4] introduced a related problem called "verifiable secret sharing." This problem is in some sense more general than ours, since the distributor of secrets, like the other participants, is not above cheating. In particular, the problem requires that the distribution of inconsistent shares be detected. All known solutions, including the best so far [3], [5], [6], rely on unproven assumptions such as the intractability of integer factorization or the existence of secure encryption schemes. Thus, they have the disadvantages mentioned previously in the discussion of signature schemes.

2. You Can Fool Some of the People All of the Time

This section shows how to modify Shamir's scheme so that the probability of undetected cheating is less than ε, for any $\varepsilon > 0$.

1. Choose any prime $p > \max((s-1)(k-1)/\varepsilon + k, n)$.
2. Choose $a_1, a_2, \ldots, a_{k-1} \in Z_p$ randomly, uniformly, and independently.
3. Let $q(x) = D + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$.
4. Choose $(x_1, x_2, \ldots, x_n)$ uniformly and randomly from among all permutations of n distinct elements from $\{1, 2, \ldots, p-1\}$. Let $D_i = (x_i, d_i)$, where $d_i = q(x_i)$.

Note that the key difference between this and Shamir's scheme occurs in step 4. The proofs of properties (a) and (b) were given by Shamir, and are sketched here for completeness.

(a) Any k participants can determine the secret uniquely by interpolation, since the points $x_1, x_2, \ldots, x_n$ are distinct.

(b) Suppose participants $i_1, i_2, \ldots, i_{k-1}$ conspire to determine the secret without consulting participant i_k. When the values of D and $x_{i_1}, x_{i_2}, \ldots, x_{i_{k-1}}$ are fixed, $q(x_{i_1}), q(x_{i_2}), \ldots, q(x_{i_{k-1}})$ are functions of the random variables $a_1, a_2, \ldots, a_{k-1}$. Using the interpolation theorem and the mutual independence of $a_1, a_2, \ldots, a_{k-1}$, it can be shown that those $k - 1$ values $q(x_{i_1}), q(x_{i_2}), \ldots, q(x_{i_{k-1}})$ are uniformly distributed and mutually independent. Hence, the secret shares $D_{i_1}, D_{i_2}, \ldots, D_{i_{k-1}}$ provide no more information about the value of D than do random numbers. (This proof is somewhat more general than Shamir's, since his assumes that D is chosen by some random process, or at least viewed that way by the conspirators.)

(c) It remains to explore the probability of deceiving another participant. It will be shown that property (c) holds even if the $k - 1$ cheaters know $q(x)$, and hence know the secret. Suppose participants $i_1, i_2, \ldots, i_{k-1}$ fabricate values $(x'_{i_1}, d'_{i_1}), (x'_{i_2}, d'_{i_2}), \ldots, (x'_{i_{k-1}}, d'_{i_{k-1}})$ to send to participant i_k. Each possible secret $D' \in \{0, 1, \ldots, s - 1\}$ defines a distinct polynomial $q_{D'}(x)$ of degree at most $k - 1$ passing through the point $(0, D')$ and the fabricated points above. If $D' \neq D$, such a polynomial $q_{D'}(x)$ can intersect $q(x)$ in at most $k - 1$ points. Participant i_k will reconstruct the incorrect secret D' only if $q_{D'}(x_{i_k}) = q(x_{i_k})$ and $D' \neq D$. Recall that x_{i_k} is a random element of $\{1, 2, \ldots, p - 1\} - \{x_{i_1}, x_{i_2}, \ldots, x_{i_{k-1}}\}$. Thus for each polynomial $q_{D'}(x)$ with $D' \neq D$ the probability that $q_{D'}(x_{i_k}) = q(x_{i_k})$ is at most $(k - 1)/(p - k)$. There are $s - 1$ legal but incorrect secrets, so the fabricated values yield $s - 1$ corresponding polynomials. Any one of these polynomials would deceive participant i_k with probability at most $(k - 1)/(p - k)$. Thus the probability of deceiving participant i_k is at most $(s - 1)(k - 1)/(p - k) < \varepsilon$.

It will now be shown that this scheme runs in expected time polynomial in k, n, $\log s$, and $\log(1/\varepsilon)$. It suffices to demonstrate that the expected time is polynomial in k, n, and $\log p$, since p may always be chosen so that $\log p$ is linear in $\log k$, $\log n$, $\log s$, and $\log(1/\varepsilon)$. A certified prime p of this magnitude can be found in expected time polynomial in $\log p$ [1]. The random choice of $a_1, a_2, \ldots, a_{k-1}$ and $(x_1, x_2, \ldots, x_n)$ can be done in expected time polynomial in k, n, and $\log p$, as can the evaluation of $q(x)$ at n points over Z_p. Finally, interpolating k points over Z_p can be done in time polynomial in k and $\log p$ [2], [8].

3. How To Keep a Secret from Cheaters

Unfortunately, although cheaters are detected with high probability, they obtain the secret while the other participants gain no information about the secret. The reader can probably imagine applications in which this would be unacceptable.

A simple solution is to augment the set $\{0, 1, \ldots, s - 1\}$ of legal values by the addition of a dummy legal value, say s, that is never used as the value of a real secret. The true secret D is now encoded as a sequence $D^{(1)}, D^{(2)}, \ldots, D^{(t)}$ where

$D^{(i)} = D$ for some i chosen randomly and uniformly, and $D^{(j)} = s$ for all $j \neq i$. Each element of this sequence is then divided into shares using probabilistically independent applications of the scheme of Section 2.

When k participants agree to pool their shares, they reconstruct $D^{(1)}, D^{(2)}, \ldots$ one at a time, until some $D^{(j)} \neq s$ is obtained. This terminates the protocol. If $D^{(j)}$ is not legal, then cheating has occurred.

Let i denote the round for which $D^{(i)} = D$; i is a random variable whose value is unknown to the cheaters. Let the cheaters choose their optimal strategy. Let e_i denote the event that the protocol does not terminate before round i and the cheaters submit fabricated shares at round i; e_i is the bad event to which the first paragraph of this section alluded. Finally, let $p(t) = \Pr(e_i)$. Then $p(t) < (1-\varepsilon)^{-1}t^{-1}$, by induction on t:

Basis ($t = 1$). $p(t) \leq 1 < (1-\varepsilon)^{-1}$.

Induction ($t > 1$). Let p_1 denote the probability with which the cheaters decide to submit fabricated shares at round 1. Let s_1 denote the event that the protocol does not terminate in round 1. Then

$$\begin{aligned} p(t) &= \Pr(i=1)\Pr(e_i \mid i=1) + \Pr(i>1)\Pr(s_1 \mid i>1)\Pr(e_i \mid i>1 \& s_1) \\ &= t^{-1}p_1 + (t-1)t^{-1}(p_1\varepsilon + (1-p_1))p(t-1) \\ &< t^{-1}(p_1 + (t-1)(p_1\varepsilon + (1-p_1))(1-\varepsilon)^{-1}(t-1)^{-1}) \\ &= (1-\varepsilon)^{-1}t^{-1}. \end{aligned}$$

Acknowledgments

We thank Don Coppersmith, Oded Goldreich, and Prabhakar Raghavan for helpful conversations.

References

[1] L. M. Adleman and M.-D. A. Huang, Recognizing primes in random polynomial time, *Proc. 19th Annual ACM Symp. Theory Comput.*, pp. 462–469, May 1987.

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[3] J. C. Benaloh, Secret sharing homomorphisms: keeping shares of a secret secret, *Advances in Cryptology—CRYPTO '86*, pp. 251–260, Lecture Notes in Computer Science, vol. 263, Springer-Verlag, Berlin, 1987.

[4] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, Verifiable secret sharing and achieving simultaneity in the presence of faults, *Proc. 26th Symp. Found. Comp. Sci.*, pp. 383–395, October 1985.

[5] P. Feldman, A practical scheme for noninteractive verifiable secret sharing, *Proc. 28th Symp. Found. Comp. Sci.*, pp. 427–437, October 1987.

[6] O. Goldreich, S. Micali, and A. Wigderson, How to prove all NP statements in zero-knowledge and a methodology of cryptographic protocol design, *Advances in Cryptology—CRYPTO '86*, pp. 171–185, Lecture Notes in Computer Science, vol. 263, Springer-Verlag, Berlin, 1987.

[7] S. Goldwasser, S. Micali, and R. L. Rivest, A "paradoxical" solution to the signature problem, *Proc. 25th Symp. Found. Comp. Sci.*, pp. 441–448, October 1984.

[8] J. D. Lipson, *Elements of Algebra and Algebraic Computing*, Addison-Wesley, Reading, MA, 1981.

[9] M. O. Rabin, Randomized Byzantine generals, *Proc. 24th Symp. Found. Comp. Sci.*, pp. 403–409, November 1983.

[10] A. Shamir, How to share a secret, *Comm. ACM*, **22**(11): 612–613, November 1979.