

How to Share a Secret with Cheaters

Martin Tompa

IBM Thomas J. Watson Research Center

P. O. Box 218

Yorktown Heights, New York 10598

Heather Woll

Department of Computer Science, FR35

University of Washington

Seattle, Washington 98195

Abstract

This paper demonstrates that Shamir's scheme ("How to share a secret", *Communications of the ACM*, vol. 22, no. 11, November 1979, 612-613) is not secure against cheating. A small modification to his scheme retains the security and efficiency of the original, is secure against cheating, and preserves the property that its security does not depend on any unproven assumptions such as the intractability of computing number-theoretic functions.

1. How to Cheat when Sharing a Secret

Shamir [7] 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" $D \in \{0, 1, \dots, s-1\}$.

Problem: Divide D into "shares" D_1, D_2, \dots, D_n such that

- knowledge of any k shares is sufficient to efficiently reconstruct D , and
- knowledge of any $k - 1$ shares provides no more information about the value of D than was known before.

This material is based in part upon work supported by the National Science Foundation under grants DCR-8301212 and DCR-8352093.

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 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 $a_1, a_2, \dots, a_{k-1} \in Z_p$ randomly, uniformly, and independently.
3. Let $q(x) = D + a_1x + a_2x^2 + \dots + a_{k-1}x^{k-1}$.
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.)

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 $\epsilon > 0$ that any $k - 1$ participants i_1, i_2, \dots, i_{k-1} can fabricate new shares $D'_{i_1}, D'_{i_2}, \dots, D'_{i_{k-1}}$ that deceive a k th participant i_k . Here, deceiving the k th participant means that, from $D'_{i_1}, D'_{i_2}, \dots, D'_{i_{k-1}}$, and D_{i_k} , the secret D' reconstructed is "legal" (i.e., $D' \in \{0, 1, \dots, s - 1\}$), 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 that the stronger version of condition (c) resulting when $\epsilon = 0$ 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 $D'_{i_1}, D'_{i_2}, \dots, D'_{i_{k-1}}$ such that the collection of shares $\{D'_{i_1}, D'_{i_2}, \dots, D'_{i_{k-1}}, D_{i_k}\}$ represents the secret D' , thus deceiving the k th 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 k th participant to detect cheating. One 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, \dots, 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 $\Delta(i_2) = \Delta(i_3) = \dots = \Delta(i_k) = 0$. 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 3 of 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 running time is polynomial in $k, n, \log s$ and $\log(1/\epsilon)$.

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 [5]). This is, in fact, exactly the solution that Rabin [6] chose when he used Shamir's scheme to solve the problem of agreement among distributed processes that might cheat. There are 2 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 [2] 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 pieces be detected. All known solutions, including the best so far [3], 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 $\epsilon > 0$.

1. Choose any prime $p > \max((s-1)(k-1)/\epsilon + k, n)$.
2. Choose a_1, a_2, \dots, a_{k-1} in Z_p randomly, uniformly, and independently.
3. Let $q(x) = D + a_1x + a_2x^2 + \dots + a_{k-1}x^{k-1}$.
4. Choose (x_1, x_2, \dots, x_n) uniformly and randomly from among all permutations of n distinct elements from $\{1, 2, \dots, p-1\}$. 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 presented here for completeness.

- (a) Any k participants can determine the secret uniquely by interpolation, since the points x_1, x_2, \dots, x_k are distinct.
- (b) Suppose participants i_1, i_2, \dots, i_{k-1} conspire to determine the secret without consulting participant i_k . When the values of D and $x_{i_1}, x_{i_2}, \dots, x_{i_{k-1}}$ are fixed, $q(x_{i_1}), q(x_{i_2}), \dots, q(x_{i_{k-1}})$ are functions of the random variables a_1, a_2, \dots, a_{k-1} . Using the interpolation theorem and the mutual independence of a_1, a_2, \dots, a_{k-1} , it is straightforward to show that those $k-1$ values $q(x_{i_1}), q(x_{i_2}), \dots, q(x_{i_{k-1}})$ are uniformly distributed and mutually inde-

pendent. Hence, the secret shares $D_{i_1}, D_{i_2}, \dots, D_{i_{k-1}}$ 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, \dots, i_{k-1} fabricate values $(x'_{i_1}, d'_{i_1}), (x'_{i_2}, d'_{i_2}), \dots, (x'_{i_{k-1}}, d'_{i_{k-1}})$ to send to participant i_k . Each possible secret $D' \in \{0, 1, \dots, s - 1\}$ 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 $\{1, 2, \dots, p - 1\} - \{x_{i_1}, x_{i_2}, \dots, x_{i_{k-1}}\}$. 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 $(s - 1)(k - 1)/(p - k) < \epsilon$.

It will now be shown that this scheme runs in expected time polynomial in $k, n, \log s$, and $\log(1/\epsilon)$. 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 $\log(1/\epsilon)$. A certified prime p of this magnitude can be found in expected time polynomial in $\log p$ [4]. The random choice of a_1, a_2, \dots, a_{k-1} and (x_1, x_2, \dots, 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 $\log p$ [1].

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 introduce 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 $D^{(1)}, D^{(2)}, \dots, D^{(t)}$ where $D^{(i)} = D$ for some i chosen randomly, and $D^{(j)} = s$ for all $j \neq i$. Each element of this sequence is then divided into shares using the scheme of section 2.

When k participants agree to pool their shares, they reconstruct $D^{(1)}, D^{(2)}, \dots$ one at a time, until some $D^{(i)} \neq s$ is obtained. If $D^{(i)}$ is not legal, then cheating has occurred. The probability of cheating on the one crucial round while going undetected at any possible earlier cheats is less than

$$t^{-1} + \epsilon t^{-1} + \epsilon^2 t^{-1} + \dots = (1 - \epsilon)^{-1} t^{-1}.$$

(This can be proven more formally by induction on t . Recall that even if the cheater suspects the secret is s , the probability of undetected cheating is at most ϵ .)

Acknowledgements

We thank Don Coppersmith for several helpful conversations. Oded Goldreich kindly provided constructive criticism on an earlier draft.

Bibliography

1. A. Aho, J. Hopcroft, and J. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
2. B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. *Proc. 26th IEEE Symp. Found. Comp. Sci.*, pages 383-395, October 1985.
3. O. Goldreich, S. Micali, and A. Wigderson. How to prove all NP statements in zero-knowledge and a methodology of cryptographic protocol design. *These Proceedings*, 1987.
4. S. Goldwasser and J. Killian. Almost all primes can be quickly certified. *Proc. 18th Annual ACM Symp. Theory Comput.*, pages 316-329, May 1986.
5. S. Goldwasser, S. Micali, and R. Rivest. A 'paradoxical' solution to the signature problem. *Proc. 25th IEEE Symp. Found. Comp. Sci.*, pages 441-448, October 1984.
6. M. Rabin. Randomized Byzantine generals. *Proc. 24th IEEE Symp. Found. Comp. Sci.*, pages 403-409, November 1983.
7. A. Shamir. How to share a secret. *Comm. ACM*, 22(11):612-613, November 1979.