How To Broadcast A Secret

Shimshon Berkovits

University of Lowell
and
The MITRE Corporation
Burlington Road
Bedford, MA 01730 USA

Abstract. A single transmitter wishes to broadcast a secret to some subset of his listeners. He does not wish to perform, for each of the intended recipients, a separate encryption either of the secret or of a single key with which to protect the secret. A general method for such a secret broadcasting scheme is proposed. It is based on "k out of n" secret sharing. An example using polynomial interpolation is presented as well as a related vector formulation.

INTRODUCTION AND HISTORY

A single transmitter (such as a key distribution center) wishes to broadcast a secret S which is an integer (and could represent a cryptographic key). However, the broadcaster does not wish the secret to be intelligible to all possible listeners. In fact, he wants only a certain, recently selected subset of subscribers to recover S. All other subscribers (and anyone else listening) should either compute nonsense or should be unable to complete the computations all together. We refer to the subscribers either as recipients or as non-recipients depending on whether they are to receive S or not. All other listeners are eavesdroppers.

Till now, the only technique available to the transmitter is the obvious one - with variations. He encrypts S in some random key R. Before he broadcasts the encrypted secret $E_R(S)$, he

D.W. Davies (Ed.): Advances in Cryptology - EUROCRYPT '91, LNCS 547, pp. 535-541, 1991.

must send R to each recipient. To do this securely, the transmitter encrypts R individually and separately in the private key encryption key of each recipient. He then communicates the result of these encryptions individually to each recipient. There are several ways he can do this as part of the same message in which he broadcasts $E_R(S)$. He reserves a field in the message for each subscriber. In the field associated with a recipient, he places R encrypted under that subscriber's private key. In all other subscriber fields, he places some null sequence. Alternatively, if there are many subscribers but only a few who are to receive S, he enters the identification of each recipient followed by R encrypted for that subscriber. In either of these variations, the transmitter is using the broadcast message to reach each separate recipient "in series."

The broadcaster can, if he prefers, communicate with all the individual recipients "in parallel." First, each subscriber is given a unique integer n_i which is larger than the encryption of any key in that subscriber's private key encryption key. The transmitter computes a single integer R' that, modulo each n_i, is congruent to the appropriate encryption of R when the n_i belongs to a recipient and is congruent to a null otherwise([3]). The integer R' is of the same size as the product of all the n_i for all subscribers. The broadcaster can reduce its size and save on transmission if he does not send nulls to the non-recipients. Then R' is of the same magnitude as the product only of the n_i belonging to recipients. However, non-recipients will have no way of knowing that the key they compute is erroneous. Only if the secret has an expected meaning or an expected format can they tell that they were not to receive the secret.

All these schemes are variations on the same theme. The transmitter, for convenience, uses a single message to send information intended for and unique to each individual recipient. A true broadcast scheme, however, is one in which the broadcast message contains the same information for each and every listener. From it, the recipients each deduce the secret and all others derive nonsense or nothing. Such schemes are possible.

A true broadcast system can be created out of any "k out of n" secret sharing scheme (e. g., [1], [4]). This is not entirely surprising since the broadcaster wishes to share a secret with each individual recipient. The remainder of this paper describes how to create such a system. The ideas are made concrete by using a system derived from Shamir's polynomial interpolation secret sharing scheme. Finally, a vector based method related to Brickell's secret sharing is presented.

TRUE BROADCAST SCHEMES

In secret sharing schemes, each participant gets a share in the secret. For a "k out of n" scheme, any k of the n participants can pool their shares and reconstruct the secret. Shamir's method, which serves as our example, encodes the secret S into the coefficients of some polynomial P of degree k - 1. For simplicity, assume that S is the constant term of P. Each of the n shares is then a distinct point on the graph of P. Obviously, any k participants can pool their shares, interpolate P and recover S.

In a secret broadcasting system, participants are given pseudoshares which are shares in an, as yet, uncreated "k out of n" system. In our example, pseudoshares are points (x_i, y_i) with distinct values x_i. The broadcaster keeps some unassigned pseudoshares for his own use in order to introduce a degree of randomness into his messages. If he ever must resend a secret to a particular set of recipients, he can create a totally different message to do so.

To broadcast to k recipients:

1. Choose $j \geq 0$.
2. Create a $k + j + 1$ out of $2k + j + 1$ secret sharing system with
    a) Secret = S
    b) Pseudoshares of recipients as real shares
    c) Pseudoshares of non-recipients must not be real shares
    d) Broadcaster includes j randomly chosen, unassigned pseudoshares.
3. Broadcast $k + j$ randomly chosen shares - all different from those used in step 2.
4. Each subscriber adds his pseudo share as a possible share to the $k + j$ shares received
    a) If that pseudoshare is an actual share, as in 2b), he recovers S
    b) If not, as in 2c), he does not recover S.

For our polynomial example:

1. Choose j points (x_i, y_i) with unassigned x_i.
2. Find a polynomial P of degree $k + j$ passing through

a) $(0, S)$
b) (x_i, y_i) for the k real recipients and the j dummy recipients belonging to the broadcaster
c) no (x_i, y_i) for any non-recipient.

3. Broadcast $k + j$ other points on the graph of P.

It is worth noting that, if all the y_i are equal to some fixed b, then Rolle's Theorem guarantees that 2c) is satisfied. By 2b), there are $k + j - 1$ intervals on which P has value b at the endpoints. Thus, there are $k + j - 1$ local extrema. Another point at which P has the value b would mean another local extreme point. But P has degree $k + j$ so it has at most $k + j - 1$ local extrema. Therefore, P cannot pass through any other pseudoshare with $y_i = b$. Of course, S cannot also equal b or P would just be a constant function. Also, each pseudoshare reduces to the value of x_i only. The broadcaster can change b daily or even from message to message but whether there is a security penalty for choosing all $y_i = b$ still needs examination.

A VECTOR BASED BROADCAST SYSTEM

The first $k + j$ equations each receiver uses in our polynomial example are the same for all participants. Only the $(k + j + 1)^{st}$ equation is unique to each subscriber. The broadcaster, who may have greater computing resources, can reduce those first $k + j$ equations and transmit only the result. This leads to a vector formulation which is related to Brickell's secret sharing scheme ([2]).

1. Pseudoshares are pairs (v_i, y_i) where the set of all vectors v_i, including the broadcaster's dummy pseudoshares form an independent set.
2. Pick a random vector $\boldsymbol{P}$ (equivalent to the coefficients of Shamir's polynomial) such that
   a) $\boldsymbol{P} \cdot v_i = y_i$ for the k pseudoshares of the recipients and the j dummy pseudoshares of the broadcaster
   b) $\boldsymbol{P} \cdot v_i \neq y_i$ for the pseudoshares of the non-recipients.
3. Pick a random vector $\boldsymbol{u}$ with $\boldsymbol{u} \cdot v_i \neq 0$ for any recipient.
4. Broadcast $\boldsymbol{u}$ and the vector $\boldsymbol{Q} = \boldsymbol{P} + S\boldsymbol{u}$.
5. Each subscriber solves $(\boldsymbol{Q} - T\boldsymbol{u}) \cdot v_i = y_i$ for T.

The last computation is

$$T = (\boldsymbol{Q} \cdot \boldsymbol{v}_i - y_i)(\boldsymbol{u} \cdot \boldsymbol{v}_i)^{-1} \tag{1}$$

Any recipient will recover $T = S$. Any non-recipient will compute a T which is different from S provided that $\boldsymbol{P} \cdot \boldsymbol{v}_i = y_i$ does not happen accidentally for a non-recipient. The broadcaster can assure that a non-recipient who does not accidentally recover S cannot complete the computation of T. He does this by choosing $\boldsymbol{u}$ with $\boldsymbol{u} \cdot \boldsymbol{v}_i = 0$ for all non-recipients. This is nice for it immediately informs the non-recipients they were not to get the secret. Unfortunately, there is an obvious computational cost to the broadcaster for choosing such a $\boldsymbol{u}$.

The choice of the random vector $\boldsymbol{u}$ with $\boldsymbol{u} \cdot \boldsymbol{v}_i \neq 0$ for any recipient is quite easy The broadcaster selects random, non-zero numbers r_i and solves $\boldsymbol{u} \cdot \boldsymbol{v}_i = r_i$ for each recipient at the same time he solves $\boldsymbol{P} \cdot \boldsymbol{v}_i = y_i$. There is an interesting variant of the scheme in which each pseudoshare consists of $(\boldsymbol{v}_i, y_i, z_i)$. The broadcaster solves, for each intended recipient, $\boldsymbol{P} \cdot \boldsymbol{v}_i = y_i$ and $\boldsymbol{u} \cdot \boldsymbol{v}_i = z_i^{-1}$. Then he need broadcasts only $\boldsymbol{Q}$ as each receiver uses his own z_i in place of the $(\boldsymbol{u} \cdot \boldsymbol{v}_i)^{-1}$ of equation (1). Of course, it no longer is useful to chose $\boldsymbol{u}$ so that $\boldsymbol{u} \cdot \boldsymbol{v}_i = 0$ for all non-recipients because everyone is using his own z_i and not computing any reciprocals.

There is another way the broadcaster can guarantee that no non-recipient can accidentally find S. As with Rolle's Theorem, he chooses all y_i equal to some fixed b. In the scheme which follows, truncation to length k means projection on the first k coordinates.

1. Pseudoshares are vectors $\boldsymbol{v}_i$ chosen so that any set of differences $\{\boldsymbol{v}_1 - \boldsymbol{v}_2, \boldsymbol{v}_1 - \boldsymbol{v}_3, \ldots, \boldsymbol{v}_1 - \boldsymbol{v}_k\}$, when truncated to length k, are independent and span a space containing no other similarly truncated $\boldsymbol{v}_1 - \boldsymbol{v}_r$.
2. If the $\boldsymbol{v}_i$ are truncated to length $k + j$, the broadcaster can find a $\boldsymbol{P}$ such that $(\boldsymbol{v}_{i_1} - \boldsymbol{v}_{i_m}) \cdot \boldsymbol{P} = 0$ for k real recipients and for j dummy recipients. There is a one dimensional solution space. Choose $\boldsymbol{P}$ in it with $\boldsymbol{v}_{i_1} \cdot \boldsymbol{P} = b$. Then $\boldsymbol{v}_{i_m} \cdot \boldsymbol{P} = b$ for all k recipients.
3. The remainder of the scheme proceeds as before with all subscribers truncating their vectors to $k + j$ components.

Theorem 1: A non-recipient with vector v_r cannot recover S.
Proof: Assume $v_r \cdot P = b$ for a non-recipient. Then $(v_{i_1} - v_r) \cdot P = 0$ and the truncated vectors $v_{i_1} - v_{i_2}, v_{i_1} - v_{i_3}, \ldots, v_{i_1} - v_{i_{k+j}}$ and $v_{i_1} - v_r$ are a basis for the entire $k + j + 1$ dimensional space. So P must be $\mathbf{0}$. But $v_{i_1} \cdot P = b$. Contradiction! Hence, $v_r \cdot P \neq b$. ♦

The next theorem suggests a way to choose the vectors v_{i_m} so that they posses the property described in Step 1 of the last scheme. The polynomial origins of the scheme become more apparent.

Theorem 2: Choose a random n-vector A. Let $v_i = (x_i, x_i^2, x_i^3, \ldots, x_i^n) + A$ with distinct x_i. Then any set of differences $\{v_1 - v_2, v_1 - v_3, \ldots, v_1 - v_k\}$, when truncated to length k, will span a space which does not contain any other $v_1 - v_r$.
Proof: Consider the Vandermonde matrix formed by $x_1, x_2, x_3 \ldots, x_k, x_r$. Since the x_i are distinct, the determinant is non-zero. But

$$\begin{vmatrix} 1 & 1 & 1 & \ldots & 1 & 1 \\ x_1 & x_2 & x_3 & \ldots & x_k & x_r \\ x_1^2 & x_2^2 & x_3^2 & \ldots & x_k^2 & x_r^2 \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ x_1^k & x_2^k & x_3^k & \ldots & x_k^k & x_r^k \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ x_1 & x_2 - x_1 & x_3 - x_1 & \ldots & x_k - x_1 & x_r - x_1 \\ x_1^2 & x_2^2 - x_1^2 & x_3^2 - x_1^2 & \ldots & x_k^2 - x_1^2 & x_r^2 - x_1^2 \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ x_1^k & x_2^k - x_1^k & x_3^k - x_1^k & \ldots & x_k^k - x_1^k & x_r^k - x_1^k \end{vmatrix}$$

Expanding the second determinant by the first row yields a k x k non-zero determinant whose columns must be linearly independent. Obviously, these columns are exactly the truncated vectors $v_1 - v_2, v_1 - v_3, \ldots, v_1 - v_k$ and $v_1 - v_r$. ♦

Although this version assures that no non-recipient can accidentally compute the secret, all non-recipients do compute something. As before, with some additional computation, the broadcaster can choose u so that $u \cdot v_i = 0$ for all non-recipients. To do this, he cannot truncate the vectors to length any less than $n - k$, the number of non-recipients.

CONCLUSION

We have shown how to convert any "k out of n" secret sharing scheme into a secret broadcasting scheme. We have also developed a specific vector based broadcast scheme which allows several variations. At one extreme, vectors are truncated to save computational effort and transmission time while, at the other, an extra matrix reduction ensures that non-recipients know they are not to get the secret because they cannot complete the required computation.

Some examination of the security of these schemes is still necessary. One can think of $\boldsymbol{Q} = \boldsymbol{P} + S\boldsymbol{u}$ and $\boldsymbol{u}$ as an encryption of the secret S with $\boldsymbol{P}$ and $\boldsymbol{u}$ chosen with somewhat randomly. However, $\boldsymbol{P}$ was not a completely random vector but is selected to satisfy certain conditions. Just how secure this encryption of S is must still be determined.

LIST OF REFERENCES

1. Blakley, G. R., "Safegaurding Cryptographic Keys," Proceedings of the. AFIPS 1979 National Computer Conference, Vol. 48, June 1979, pp. 313-317.

2. Brickell, E. F., "Some Ideal Secret Sharing Schemes," Third Carbondale Combinatorics Conference, Carbondale, IL, J. Combinatorial Mathematics and Combinatorial Computing, Vol. 6, 1989, pp. 105-113.

3. Chiou, G. C. and W. C. Chen, "Secure Broadcasting Using the Secure Lock," IEEE Transactions on Software Engineering, Vol. SE-15, No. 8, Aug. 1989, pp. 929-934.

4. Shamir, A., "How to Share a Secret," Communications of the ACM, Vol. 22, No. 11, November 1979, pp. 612-613.