





	PAGE	BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO	. 3. RECIPIENT'S CATALOG NUMBER
MTT/1 CS/TM-124		
4. TITLE (and Subtitie)		5. TYPE OF REPORT & PERIOD COVER
7		
How to Share a Secret .		
1	ALL.	C. PERFORMING ORC. REPORT NUMBER
	(14	MIT/LCS/TM-134
7. AUTHOR(s)	-	8. CONTRACT OR GRANT NUMBER(s)
Justin	(1)	1 N00014-76-C-0366
Adi Shamir	(1)	L'appril 10 c psoo
	~	
MIT / I aboratory for Computer Color	/	AREA & WORK UNIT NUMBERS
545 Technology Square	ce	
Cambridge MA 02139		
Cambridge, FA 02155		12 000000000000000000000000000000000000
ONR/Department of the Navy		Apr (1 1070
Information Systems Program	U	12 MMDED 05 04005
Arlington, VA 22217		9
14. MONITORING AGENCY NAME & ADDRESS(II differen	t from Controlling Office)	15. SECURITY CLASS. (of this report)
(1) YY, D. /		Unclassified
		15. DECLASSIFICATION DOWNGRADING
		SCHEDULE
Approved for public release; distr	ibution unlimite	ed
		Papart)
17. DISTRIBUTION STATEMENT (of the abstract entered	in Block 20, if different fra	om Report)
17. DISTRIBUTION STATEMENT (of the obstract entered	in Block 20, if different fro	om Report)
17. DISTRIBUTION STATEMENT (of the abatract entered	In Block 20, 11 different in	om Report)
17. DISTRIBUTION STATEMENT (of the aboutact entered	in Block 20, 11 dilloroni Iro MEMOIS	om Report)
17. DISTRIBUTION STATEMENT (of the ebetract entered 9) Technical 18. SUPPLEMENTARY NOTES	In Block 20, 11 different fro MEMDIS	om Report)
17. DISTRIBUTION STATEMENT (of the abatract entered 9 Technical 18. SUPPLEMENTARY NOTES	In Block 20, 11 different In MEMOIS	om Report)
17. DISTRIBUTION STATEMENT (of the aboutact entered 9 Technical 18. SUPPLEMENTARY NOTES	In Block 20, 11 dilloroni Iro	om Report)
17. DISTRIBUTION STATEMENT (of the aboutact entered 9 Technical 18. SUPPLEMENTARY NOTES	In Block 20, 11 dilloroni In Memolis	om Report)
17. DISTRIBUTION STATEMENT (of the obstract entered () Technical 18. SUPPLEMENTARY NOTES	in Block 20, 11 different fro MEMDIS	om Report)
17. DISTRIBUTION STATEMENT (of the abatract entered DTethnical 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary an Cryptography	In Block 20, 11 different fro MEMDIS Id identify by block number	om Report)
<ul> <li>DISTRIBUTION STATEMENT (of the abstract entered</li></ul>	in Block 20, 11 different fro MEMDIS id identify by block number	om Report)
17. DISTRIBUTION STATEMENT (of the abstract entered 9. Technical 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and Cryptography key management interpolation	In Block 20, 11 different fro MEMDIS In Identify by block number	om Report)
17. DISTRIBUTION STATEMENT (of the abstract entered 9. Technical 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and Cryptography key management interpolation	in Block 20, 11 different fro MEMDIS In Identify by block number	om Report)
17. DISTRIBUTION STATEMENT (of the abatract entered DTechnical 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and Cryptography key management interpolation	in Block 20, 11 different fro MCMDIS	om Report)
17. DISTRIBUTION STATEMENT (of the ebetract entered DTechnical 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and Cryptography key management interpolation 20. ABSTRACT (Continue on reverse side if necessary and	in Block 20, 11 different fro M. C. M. O. J. d identify by block number d identify by block number)	om Report)
17. DISTRIBUTION STATEMENT (of the abatract entered () Technical 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and Cryptography key management interpolation 20. ABSTRACT (Continue on reverse side if necessary and In this paper we show how to	In Block 20, 11 different fro M. C.M. Dis d identify by block number, d identify by block number, divide data D in	to n pieces in such a way t
<ul> <li>17. DISTRIBUTION STATEMENT (of the obstract entered Description of the obstract entered 18. SUPPLEMENTARY NOTES</li> <li>19. KEY WORDS (Continue on reverse side if necessary and Cryptography key management interpolation</li> <li>20. ABSTRACT (Continue on reverse side if necessary and In this paper we show how to D is easily reconstructable from a</li> </ul>	In Block 20, 11 different fro MCMDIS d identify by block number divide data D in ny k pieces, but	to n pieces in such a way t even complete knowledge of
<ul> <li>17. DISTRIBUTION STATEMENT (of the abstract entered Dechnical Technical</li> <li>18. SUPPLEMENTARY NOTES</li> <li>19. KEY WORDS (Continue on reverse side if necessary and Cryptography key management interpolation</li> <li>20. ABSTRACT (Continue on reverse side if necessary and In this paper we show how to D is easily reconstructable from a k - 1 pieces reveals absolutely no</li> </ul>	In Block 20, 11 different fro MCMDIS d identify by block number d identify by block number divide data D in ny k pieces, but information abo	to n pieces in such a way t even complete knowledge of ut D. This technique enable
17. DISTRIBUTION STATEMENT (of the abstract entered () Technical 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and Cryptography key management interpolation 20. ABSTRACT (Continue on reverse side if necessary and In this paper we show how to D is easily reconstructable from a k - 1 pieces reveals absolutely no the construction of robust key man	In Block 20, 11 different fro M. C.M. Dis differentify by block number divide data D in ny k pieces, but information abo agement schemes	to n pieces in such a way t even complete knowledge of ut D. This technique enable for cryptographic systems
17. DISTRIBUTION STATEMENT (of the abstract entered Dechnical Technical 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and Cryptography key management interpolation 20. ABSTRACT (Continue on reverse side if necessary and In this paper we show how to D is easily reconstructable from a k - 1 pieces reveals absolutely no the construction of robust key man that can function securely and rel	In Block 20, 11 different fro MEMDIS d identify by block number d identify by block number divide data D in ny k pieces, but information abo agement schemes iably even when	to n pieces in such a way t even complete knowledge of ut D. This technique enable for cryptographic systems misfortunes destroy half the
<ul> <li>DISTRIBUTION STATEMENT (of the abstract entered</li></ul>	In Block 20, 11 different fro MEMOIS d identify by block number d identify by block number divide data D in ny k pieces, but information abo agement schemes iably even when e all but one of	to n pieces in such a way t even complete knowledge of ut D. This technique enable for cryptographic systems misfortunes destroy half the the remaining pieces.
<ul> <li>17. DISTRIBUTION STATEMENT (of the abstract entered</li></ul>	In Block 20, 11 different for MCMOIS d identify by block number divide data D in ny k pieces, but information abo agement schemes iably even when e all but one of	to n pieces in such a way t even complete knowledge of ut D. This technique enable for cryptographic systems misfortunes destroy half the the remaining pieces.
<ul> <li>17. DISTRIBUTION STATEMENT (of the obstract entered Decknical Decknical)</li> <li>18. SUPPLEMENTARY NOTES</li> <li>19. KEY WORDS (Continue on reverse side if necessary and Cryptography key management interpolation</li> <li>10. ABSTRACT (Continue on reverse side if necessary and In this paper we show how to D is easily reconstructable from a k - 1 pieces reveals absolutely no the construction of robust key man that can function securely and rel pieces and security breaches exposition</li> </ul>	In Block 20, 11 different free M. C. M. O. J. d identify by block number, divide data D in ny k pieces, but information abo agement schemes iably even when e all but one of	to n pieces in such a way to even complete knowledge of ut D. This technique enable for cryptographic systems misfortunes destroy half the the remaining pieces.
<ul> <li>DISTRIBUTION STATEMENT (of the abstract entered</li></ul>	In Block 20, 11 different for MCMOIS a identify by block number divide data D in ny k pieces, but information abo agement schemes iably even when e all but one of .ETE	to n pieces in such a way t even complete knowledge of ut D. This technique enable for cryptographic systems misfortunes destroy half the the remaining pieces.
<ul> <li>DISTRIBUTION STATEMENT (of the abstract entered</li></ul>	In Block 20, 11 different fro M. C.M. Dij d identify by block number divide data D in ny k pieces, but information abo agement schemes iably even when e all but one of .ETE SECURITY CL	to n pieces in such a way to even complete knowledge of ut D. This technique enable for cryptographic systems misfortunes destroy half the the remaining pieces.

SECURITY CLASSIFICATION OF THIS PAGE(When Date Entered)

ę

-the

#### SECURITY CLASSIFICATION OF THIS PAGE(When Date Entered)

Se Statest

#### MIT/LCS/TM-134

HOW TO SHARE A SECRET

ADI SHAMIR

**APRIL 1979** 

Access	ion For	
NTIS	Guldal	1
DDC TI	B	
Justii	ication	<u> </u>
By		
Distri	hation/	
/vri)	171 A C	cdes
	A:ailand,	'or
Dist	special	
A		
H		

This research was supported by the Office of Naval Research under Contract No. N00014-76-C-0366.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY LABORATORY FOR COMPUTER SCIENCE

MASSACHUSETTS 02139

79 05 29 045

CAMBRIDGE

#### How to Share a Secret

Adi Shamir

Department of Mathematics Massachusetts Institute of Technology Cambridge, Massachusetts 02139

#### April, 1979

<u>Abstract</u>: In this paper we show how to divide data D into n pieces in such a way that D is easily reconstructable from any k pieces, but even complete knowledge of k - 1 pieces reveals absolutely no information about D. This technique enables the construction of robust key management schemes for cryptographic systems that can function securely and reliably even when misfortunes destroy half the pieces and security breaches expose all but one of the remaining pieces.

Key words: Cryptography, key management, interpolation.

This research was supported by the Office of Naval Research under Contract No. N00014-76-C-0366.

#### 1. Introduction.

In [1], Liu considers the following problem:

"Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. What is the smallest number of locks needed? What is the smallest number of keys to the locks each scientist must carry?"

It is not hard to show that the minimal solution uses 462 locks and 252 keys per scientist. These numbers are clearly impractical, and they become exponentially worse when the numbers of scientists increases.

In this paper we generalize the problem to one in which the secret is some data D (e.g., the safe combination) and in which non-mechanical solutions (which manipulate this data) are also allowed. Our goal is to divide D into n pieces  $D_1, \ldots, D_n$  in such a way that:

- knowledge of any k or more D<sub>i</sub> pieces makes D easily computable;
- (2) knowledge of any k 1 or fewer D<sub>i</sub> pieces leaves D completely undetermined (in the sense that all its possible values are equally likely).

Such a scheme is called a (k,n) threshold scheme.

Efficient threshold schemes can be very helpful in the management of cryptographic keys. In order to protect data we can encrypt it, but in order to protect the encryption key we need a different method (further encryptions change the problem rather than solve it). The most secure key management scheme keeps the key in a single, well-guarded location (a computer, a human brain, or a safe). This scheme is highly

-2-

unreliable since a single misfortune (a computer breakdown, sudden death or sabotage) can make the information inaccessible. An obvious solution is to store multiple copies of the key at different locations, but this increases the danger of security breaches (computer penetration, betrayal or human errors). By using a (k,n) threshold scheme with n = 2k - 1we get a very robust key management scheme: we can recover the original key even when  $\lfloor \frac{n}{2} \rfloor = k - 1$  of the n pieces are destroyed, but our opponents cannot reconstruct the key even when security breaches expose  $\lfloor \frac{n}{2} \rfloor = k - 1$ of the remaining k pieces.

In other applications the tradeoff is not between secrecy and reliability, but between safety and convenience of use. Consider, for example, a company that digitally signs all its checks (see RSA [2]). If each executive is given a copy of the company's secret signature key, the system is convenient but easy to misuse. If the cooperation of all the company's executives is necessary in order to sign each check, the system is safe but inconvenient. The standard solution requires at least three signatures per check, and it is easy to implement with a (3,n) threshold scheme. Each executive is given a small magnetic card with one  $D_i$  piece, and the company's signature generating device accepts any three of them in order to generate (and later destroy) a temporary copy of the actual signature key D. The device does not contain any secret information and thus it need not be tamper-proof. An unfaithful executive must have at least two accomplices in order to forge the company's signature in this scheme.

Threshold schemes are ideally suited to applications in which a group

-3-

of mutually suspicious individuals with conflicting interests must cooperate. Ideally we would like the cooperation to be based on mutual consent, but the veto power this mechanism gives to each member can paralyze the activities of the group. By properly choosing the k and n parameters we can give any sufficiently large majority the authority to take some action while giving any sufficiently large minority the power to block it.

### 2. A simple (k,n) threshold scheme.

Our scheme is based on polynomial<sup>\*</sup> interpolation: given k points in the 2-dimensional plane  $(x_1, y_1), \ldots, (x_k, y_k)$  with distinct  $x_i$ 's, there is one and only one polynomial q(x) of degree k - 1 such that  $q(x_i) = y_i$ for all i. Without loss of generality, we can assume that the data D is (or can be made) a number. To divide it into pieces  $D_i$ , we pick a random k-1 degree polynomial  $q(x) = a_0 + a_1x + \ldots + a_{k-1}x^{k-1}$  in which  $a_0 = D$ , and evaluate:

$$D_1 = q(1), \ldots, D_i = q(i), \ldots, D_n = q(n)$$

Given any subset of k of these  $D_i$  values (together with their identifying indices), we can find the coefficients of q(x) by interpolation, and then evaluate D = q(0). Knowledge of just k - 1 of these values, on the other hand, does not suffice in order to calcuate D.

To make this claim more precise, we use modular arithmetic instead of real arithmetic. The set of integers modulo a prime number p forms a

The polynomials can be replaced by any other collection of functions which are easy to evaluate and to interpolate.

field, in which interpolation is possible. Given an integer valued data D, we pick a prime p which is bigger than both D and n. The coefficients  $a_1, \ldots, a_{k-1}$  in q(x) are randomly chosen from a uniform distribution over the integers in [0,p), and the values  $D_1, \ldots, D_n$  are computed modulo p.

Let us now assume that k - 1 of these n pieces are revealed to an opponent. For each candidate value D' in [0,p) he can construct one and only one polynomial q'(x) of degree k - 1 such that q'(0) = D' and q'(i) = D<sub>i</sub> for the k - 1 given arguments. By construction, these p possible polynomials are equally likely, and thus there is absolutely nothing the opponent can deduce about the real value of D.

Efficient  $0(n \log^2 n)$  algorithms for polynomial evaluation and interpolation are discussed in [3] and [4], but even the straightforward quadratic algorithms are fast enough for practical key management schemes. If the number D is long, it is advisable to break it into shorter blocks of bits (which are handled separately) in order to avoid multi-precision arithmetic operations. The blocks cannot be arbitrarily short, since the smallest usable value of p is n+1 (there must be at least n+1 distinct arguments in [0,p) to evaluate q(x) at). However, this is not a severe limitation since sixteen bit modulus (which can be handled by a cheap sixteen bit arithmetic unit) suffices for applications with up to 64,000 D<sub>i</sub> pieces.

Some of the useful properties of this (k,n) threshold scheme (when compared to the mechanical locks and keys solutions) are:

(1) The size of each piece does not exceed the size of the original data.

-5-

(2) When k is kept fixed,  $D_i$  pieces can be dynamically added or deleted (e.g., when executives join or leave the company) without affecting the other  $D_i$  pieces.

-6-

- (3) It is easy to change the  $D_i$  pieces without changing the original data D all we need is a new polynomial q(x) with the same free term. A frequent change of this type can greatly enhance security since the pieces exposed by security breaches cannot be accumulated unless all of them are values of the same edition of the q(x) polynomial.
- (4) By using tuples of polynomial values as  $D_i$  pieces, we can get a hierarchical scheme in which the number of pieces needed to determine D depends on their importance. For example, if we give the company's president three values of q(x), each vice-president two values of q(x), and each executive one value of q(x), then a (3,n) threshold scheme enables checks to be signed either by any three executives, or by any two executives one of which is a vice-president, or by the president alone.

# **Bibliography:**

- C. L. Liu, "Introduction to Combinatorial Mathematics", McGraw-Hill, 1968.
- [2] R. Rivest, A. Shamir and L. Adleman, "A Method For Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, Vol. 21, No. 2, February 1978.
- [3] A. Aho, J. Hopcroft and J. Ullman, "The Design and Analysis of Computer Algorithms", Addison-Wesley, 1974.
- [4] D. Knuth, "The Art of Computer Programming", Vol. 2, Addison-Wesley, 1969.

## OFFICIAL DISTRIBUTION LIST

Defense Documentation Center Cameron Station Alexandria, VA 22314	r 12 copies	Naval Research Laboratory Technical Information Division Washington, D. C. 20375		
		Att: Code 2627	6	copies
Office of Naval Research Arlington, VA 22217		Dr. A. L. Slafkosky		
Att: Information Sys.Program Code 437	n 2 copies	Scientific Advisor Commandant of the Marine Corps (Code RD-1) Washington, D. C. 20380	1	CODY
Office of Naval Research Arlington, VA 22217			-	2017)
Att: Code 200	1 сору	Naval Ocean Systems Center Advanced Software Technology Divisi Code 5200	on	
Office of Naval Research Arlington, VA 22217		San Diego, CA 92152	1	cory
Att: Code 455	1 сору	Mr. E. H. Gleissner Naval Ship Research & Development C Computation & Mathematics Departmen	tr t	
Arlington, VA 22217		Bethesda, MD 20084	1	сору
Att: Code 458	1 сору	Captain Grace M. Hopper (008) Naval Data Automation Command		
Office of Naval Research Branch Office/Boston Bldg. 114, Section D		Washington Navy Yard Building 166 Washington, D. C. 20374	1	сору
Boston, MA 02210	1 сору	Mr. William O. Pateat Chief of Naval Operations (OP-160)		
Office of Naval Research Branch Office/Chicago 536 South Clark Street		Arlington, VA 20350	1	сору
Chicago, IL 60605	1 сору	Professor Omar Wing Columbia University in the city		
Office of Naval Research Branch Office/Pasadena 1030 Fast Green Street		Department of Electrical Engineerin & Computer Science	g	
Pasadena, CA 91106	1 сору	New York, N. Y. 10027	1	сору

Director National Security Agency Fort George G. Meade, MD 20755

Att: Mr. Glick

1 сору