Working Paper Series
ISSN 1170-487X

Chinks in the Armor of Public Key Cryptosystems

by William J. Wilson

Working Paper 94/3

March, 1994

Chinks in the Armor of Public Key Cryptosystems

William J. Wilson

1239 Blevins Gap Road SE,
Huntsville AL 35802 USA.
E-mail address: WARLOCK@acm.org

March 2, 1994

Abstract

Potential weaknesses in public key cryptosystem design and use are identified with emphasis on a particular vulnerability resulting from the encryption of ordinary natural language plaintext. This weakness occurs when an insufficiently long block length is used to encrypt low entropy natural language and ordinary numeric plaintext rendered in 8-bit character sets such as ASCII or EBCDIC. A computer-assisted, semi-exhaustive chosen plaintext attack is defined and shown to defeat such systems. Several methods are suggested for thwarting such attacks by maximizing input plaintext entropy prior to encryption.

Keywords Public key cryptosystems. block ciphers, message entropy, unicity distance, exhaustive enumeration, n-grams, ASCII and EBCDIC character sets.

1 Introduction

Although substantial information has not yet emerged in the open literature on general attack strategies against public key (PK) cryptosystems – and by implication, steps to be taken in their defense – the outline of certain attack strategies is gradually becoming clearer. This paper identifies potential weaknesses in block ciphers in general and in PK cryptosystem design and use in particular, outlines an successful attack strategy based on one of these weaknesses in PK systems and suggests design and use considerations for thwarting such attacks.

It has been observed since the advent of PK systems that they must be impervious to ordinary chosen plaintext attack since unlimited plaintext/ciphertext pairs can be freely created by any public key holder in spite of the presently unclear relationship of unicity distance [1] [2] [3] to public key systems. If the security of PK systems is to be upheld, such pairings must not facilitate construction of private keys or alternate workable private keys. At present, the many implications of a virtually unlimited capability for plaintext/ciphertext pair generation in the hands of a well-endowed adversary remain unclear.

In the past, statistical analysis techniques and libraries of common short n-grams served as two of the more powerful analytical weapons directed against polyalphabetic private key systems. More recently, in the cryptanalysis of private key block ciphers, computer-assisted attacks such as Hellman's proposed exhaustive enumeration of DES keyspace [4] loom each day as an ever more viable attack option against any private key block cipher where a paucity of keyspace invites such an attack. Unfortunately, the Hellman attack has ominous implications for PK system users as well. A reformulation of his brute force method can be used to mount a successful computer-assisted, semi-exhaustive chosen plaintext assault against PK systems of "insufficient block length" where such systems encrypt ordinary natural language plaintext represented in standard 8-bit ASCII or EBCDIC character sets.

Instead of exhausting key possibilities as in the Hellman attack, the adversary addresses the significantly smaller task of creating a semi-exhaustive library of the more common and likely plaintext n-grams of system block length conveniently encrypted by the ubiquitous public key.

Assume, for example, a target public key system of block length 64 bits or 8 bytes. The adversary first constructs a library of n-grams consisting of contextually likely 8-byte plaintext terms and term fragments. On receipt of the public key, he encrypts each n-gram in the library storing the resulting plaintext/ciphertext pairs by ciphertext value in a manner facilitating rapid retrieval, e.g. randomizing, hierarchical indexing, binary search, B-trees, etc. With such a database, an adversary can tell immediately whether or not a corresponding plaintext block already exists for a given ciphertext block.

Suppose the target context is submarine warfare. Encrypted forms of 8-byte text fragments such as bSUBMARI, SUBMARIN, UBMARINE, BMARINES, BMARINEb, etc. would appear frequently in plaintext input and would certainly reside in his library. Other more general but common and restricted terms and format conventions favored by military and formal communicators would also be included such as: date/time and address blocks, place names, person and item names and designators, geographic coordinates, recurring stock phrases such as ORDERbOFbBATTLE and the like, with the result that an intercepted message could be partially decrypted and immediately displayed with a substantial number of its decrypted segments *in situ*. In an interactive manner, and armed with other information, the cryptanalyst could rapidly narrow the likelihood of remaining ciphertext blocks by contextual analysis and by test encrypting likely plaintexts not in the library. The rapid convergence of such a technique can be adduced from the popular TV gameshow "Wheel of Fortune" where a small natural language phrase is guessed a letter at a time (for all occurrences) until, at a certain point and substantially before all letters are guessed, only one phrase suffices - a phenomenon exemplifying a kind of natural language unicity point.

It is conjectured that a library yielding an initial hit rate as low as 35% evenly distributed over a message text suffices to insure the rapid decryption of most remaining ciphertext blocks by the method suggested above.

Unfortunately, natural language is not the only plaintext offender. A lesser weakness occurs in the representation of numeric values (as might be found in electronic funds transfer messages) with ASCII numerals demarcated by ASCII blanks instead of a denser representation such as fixed point numbers. To exploit this weakness, an adversary could in similar manner construct a sublibrary of 8-byte ASCII numbers and number fragments as opposed to the more daunting task of encrypting and storing the full range of numbers possible in fixed point representation. Various combinations with interspersed decimal points and commas could also be included as needed noting that such symbols decrease the overall entropy of numeric plaintext and facilitate the construction of a sublibrary of numeral n-grams. Alternatively, a recursive definition of such number forms could be selectively and interactively invoked and encrypted when needed.

It is sometimes convenient to regard block ciphers as simple substitution ciphers employing "letters" of a very large alphabet to spell a secret message which is later transliterated in a one-to-many fashion to a normal alphabet by the decryption process. The innate low entropy of natural language plaintext further exacerbated by ASCII representations, insures that relatively few "letters" of such large alphabets will ever appear under a given key in any message ciphertext unless deliberately injected as nulls or filler. One need only consider for a moment the endless anomalous plaintexts whose corresponding ciphertext letters would never occur to realize that the universe of valid ciphertext "letters" is indeed a small fraction of all possibles. At the same time, certain ciphertext "letters" would occur with the same telltale frequency and distribution as their plaintext brethren - a kind of mammoth ETAOIN SHRDLU of the product cipher world. It is largely this phenomenon that makes an n-gram assault possible.

Block ciphers such as DES or RSA often suffer from a lesser weakness resulting from the direct positional correspondence between plaintext blocks and ciphertext blocks. In formally structured messages this relationship facilitates the identification of ciphertext versions of key plaintext data elements such as time stamps and addresses allowing an adversary to focus on such data elements even with longer block lengths. One need only recall the well-known weakness induced in the ciphertext productions of the famous Japanese Purple Code in WWII by the deployment of predictable honorifics at message beginnings to realize how such correspondences assisted in the recreation of their private cipher keys. Finally, ciphertext blocks produced by product ciphers, unlike earlier cipher productions, tend to be independently generated and "hermetically insulated" from other blocks allowing them to be attacked individually and in parallel unless chaining techniques are used.

2 Feasibility of n-Gram Attack for Block Length 64-Bits

The feasibility of an n-gram attack on a PK cryptosystem with a 64-bit block length is demonstrated below by showing how to construct a semi-exhaustive 8-gram plaintext library and demonstrating that its magnitude lies easily within the capabilities of readily available computing resources and that such a library suffices to decrypt the large majority of ciphertext of such cryptosystems.

A relatively exhaustive source of plaintext terms for constructing n-grams for the example model was found in the ingenious multipurpose PC computer program WORDS! and its dictionary of over 112,000 terms (all residing on a single floppy) devised by Huntsville mathematician Jack H. Allen to solve substitution ciphers, construct anagrams, and to create and solve crossword puzzles. As a result, his dictionary (which produces "indicatory" as a prophetic anagram of itself) contains needed plurals, past tenses,

Item Size	Allen Words	Reduced Words	Reduced Prefixes	Reduced Suffixes	Words+ Prefixes	Words+ Suffixes
1	26	2	26	26	26	26
2	427	51	232	252	283	303
3	2327	879	888	856	1767	1735
4	6812	3409	5896	4962	9305	8371
5	10680	5767	13683	11297	19450	17064
6	14302	8152	17799	15711	25951	23863
7	16981	11865	20805	19422	32670	31287
8	16950	11858	16707	14566	28565	26426
> 8	43604	32866				
Totals	112109	74849				

Table 1: Allen WORDS! Dictionary Statistics.

gerunds, acronyms and abbreviations, as well as common foreign terms. Totals by word length from Allen's dictionary are shown in Table 1.

A clearly better way to construct an n-gram library in actual practice – again computer assisted – would be to secure representative samples of target plaintext and to scan them incrementally left-to-right a character at a time with a "8-byte sliding window" storing each new window value occurrence or posting already encountered values in an associated "occurrence field" to assist later purging of many one-of-a-kind 8-grams. With this method, the defining syntax for acceptable n-grams would, of course, have to be constrained to avoid an unmanageable overproduction of the many n-gram permutations of upper-case/lower-case letters, punctuation symbols, multiple blanks, and the like. However constructed, an n-gram plaintext library would constitute a valuable cryptanalytic resource that could be finely tuned for each target environment and interactively updated by the analysts to reflect changing target syntax. In one sense, constructing and fine-tuning an n-gram library is essentially a one-time task – not unlike the pre-computation proposed for Hellman's assault on DES.

It is well known to information retrieval specialists engaged in auto-indexing researches that 15 or so natural language terms constitute approximately half of all natural language text. The well-known 80/20 rule seems to apply also implying that a larger but relatively small cadre of hard-working terms (substitute 8-grams) accomplish 80% of most textual work. Accordingly, assembling the basic "hard-core" of an n-gram library is the most important task. Fortunately, it is the easiest to achieve since it is composed of the most ordinary and frequent of terms.

3 Determining the Task Magnitude of an n-Gram Assault Against a PK Cryptosystem Employing a 64-Bit Block Size

Determining the task magnitude of an n-gram assault against a 64-bit block PK cryptosystem reduces in large measure to the simpler task of determining how large and exhaustive such a library must be to achieve a specified hit rate. Since the example given here does not employ the better tactic of hoisting the target ciphertext on its own linguistic petard by creating an empirical library as mentioned above, it is necessary that the library be otherwise as comprehensive as possible to insure that as few terms as possible are omitted. Fortunately, the Allen dictionary lends special credence to this approach.

Determining the size of the required 8-gram library was a three-step process. The first step consisted of defining a set of block size plaintext syntax templates shown in Table 2. consisting of every lexically reasonable combination of plaintext alphabetic characters (*) with up to three interspersed noncontiguous blanks (b). In view of the scope of the Allen dictionary, this combination was initially judged capable of achieving a hit rate of 50% in spite of the categorical exclusion of plaintext blocks not satisfying this syntax, i.e. those containing commas, periods, multiple blanks and other punctuation symbols. This step yielded 46 templates.

Since the Allen dictionary contains a wealth of words and abbreviations dear to crossword puzzle enthusiasts but eminently unlikely to occur in any plaintext, such terms were purged in the interest of overall economy. These deletions included archaic and esoteric terms, alternate spellings, rare biological and medical terms, and the like. Statistics for the resulting reduced dictionary used to construct the n-grams used in the attack are also shown in Table 1.

Not only is it necessary to identify words (i.e. n-lets where n is the word length), but it is also "prefixes" and "suffixes" ranging in size from one to eight characters as well as eight character "infixes" since the

********	= 90 327	*b*****b	= 149 942	b****b*b	= 6 818
b*******	= 32 670	*b****b*	= 2 304 484	b***b**b	= 44 829
*b******	= 674 726	*b***b**	= 6 467 682	b**b***b	= 44 829
b***	= 5 893 350	*b**b***	= 2 343 042	b*b****b	= 6 818
b*	= 16 144 175	*b*b****	= 483 860	b***b*b*	= 45 708
****b***	= 14 791 557	**b****b	= 1 032 927	b**b**b*	= 67 626
*****b**	= 30 152 088	**b***b*	= 6 924 762	b*b***b*	= 45 708
******b*	= 620 438	**b**b**	= 4 373 199	b**b*b**	= 28 866
*******b	= 31 287	**b*b***	= 1 070 802	b*b**b**	= 28 866
b******b	= 8 152	***b***b	= 1 525 065	*b***b*b	= 45 708
b*****b*	= 149 942	***b**b*	= 2 300 610	*b**b**b	= 67 626
b****b**	= 964 747	***b*b**	= 982 010	*b*b***b	= 45 708
b***b***	= 1 553 193	****b**b	= 426 921	*b**b*b*	= 68 952
b**b****	= 474 555	****b*b*	= 435 292	*b*b**b*	= 68 952
b*b*****	= 38 900	*****b*b	= 34 128	**b**b*b	= 30 906
				b*bb	= 30 906
SUBTOTALS	71 620 107		30 854 726		678 826
GRAND TOTAL	103 153 659				

Table 2: Plaintext Templates and their Resulting Populations.

executioners impartial blocking axe may fall anywhere on a plaintext string. In the context of this paper, the term "prefix" designates the beginning letters of a word while the term "suffix" designates ending letters. Infixes denote a contiguous string of interior characters in words larger than 9 characters. Prefixes are denoted by the notation n-Px where n specifies the size. In like manner, suffixes are denoted by the notation n-Sx. Totals for prefixes and suffixes computed from the reduced Allen dictionary are also shown in Table 1.

The second step in the process consisted of computing the population of each of the 46 templates using the reduced Allen dictionary. No attempt was made to exclude semantically and syntactically impossible combinations in computing these populations – a fact which errs on the side of a dictionary larger than one that would be produced empirically by sample text scanning. The final step of the process consisted simply of summing the various template totals resulting in an overall grand total of roughly 103 million n-grams. An n-gram database of this magnitude comprised of relatively short records (e.g. a 24 byte record containing the plaintext term or fragment, its ciphertext version, and 64 bits (8 bytes) reserved for various counts) would require approximately 2.4 billion bytes – a commonplace occurrence in today's database environments.

As a feasibility test of the foregoing method, the abstract paragraph of this paper was attacked as described with the result that 83% of its text was immediately decrypted. Blocks not immediately decrypted are shown below with the reason for failure:

n-Gram Not Found	Reason
Potentia	Upper case (UC) letter
intext.	Embedded period
This we	UC letter
n 8-bit	Numeral and embedded dash
uch as A	UC letter
SCII or	UC letters
EBCDIC.	UC letters and embedded period
A compu	UC letter
ter-assi	Embedded dash
sted, se	Embedded comma
mi-exhau	Embedded dash
s. Seve	Embedded period, multiple blanks, UC letter
ption.	Embedded period and multiple blanks (a typical end of message "sentinel")

A second example shown below taken from "High-Pressure Hormone" by Kathy A. Fackelman in the Dec. 1, 1990 issue of *Science News* yielded an initial hit rate of 76%.

"Members of certain East African tribes gather seeds of the tropical vine *Strophanthus gratus* and extract a lethal poison to smear on their arrow tips. Some accounts describe murderers slathering the poison on a prickly fruit, then placing the fruit on a jungle path minutes before their barefoot victim arrives."

4 Thwarting Semi-Exhaustive Chosen Plaintext Attack

Since an n-gram assault – as well as certain other analytical attacks which might be mounted against any product cipher encrypting natural language plaintext – exploits the innate low entropy of natural language plaintext, it follows that such attacks can be defeated outright by maximizing input entropy. Ideally, input plaintext block values should range over all possible values and all values should be equiprobable. Such a condition obtains naturally in many security contexts such as the encryption of digitized analog signals resulting from telephone speech but not with ordinary lexical input data and must be artificially induced.

A simple high-speed method for maximizing the entropy of any block cipher plaintext input can be achieved by an "m out of n" block scrambling technique consisting of Exclusively-ORing (XORing) selected plaintext blocks with other blocks prior to encryption and undoing these operations after decryption. As an example, let the series $A, B, C, D, E, F, \ldots$ designate a series of plaintext message blocks. Four of these blocks could be selected three at a time and XORed (where $\oplus$ designates the XOR operation) to create the following blocks values: $A \oplus B \oplus C, A \oplus B \oplus D, A \oplus C \oplus D, B \oplus C \oplus D$ which, in turn, would be delivered to the encryption module. After decryption, the resulting blocks would be XOR'd three at a time to recover the original plaintext blocks as follows:

$$(A \oplus B \oplus C) \oplus (A \oplus B \oplus D) \oplus (A \oplus C \oplus D) = A$$

$$(A \oplus B \oplus C) \oplus (A \oplus B \oplus C) \oplus (B \oplus C \oplus D) = B$$

$$(A \oplus B \oplus C) \oplus (A \oplus C \oplus D) \oplus (B \oplus C \oplus D) = C$$

$$(A \oplus B \oplus D) \oplus (A \oplus C \oplus D) \oplus (B \oplus C \oplus D) = D$$

The process would be repeated for each successive 4-block group. Many variations of this "m out of n" block scrambling technique (where m is odd and $n = m+1$) are possible although the "3 out of 4" option operates over the smallest possible set of blocks and yields a reasonably high entropy at relatively small cost.

Another tactic that may be employed at a small cost in bandwidth derives from the venerable, one-time key method of Vernam [7]. Let V represent a randomly generated bit vector of system block length. The following chained sequence is created and encrypted: $V, V \oplus A, V \oplus A \oplus B, V \oplus A \oplus B \oplus C, \ldots$ for as many blocks as desired. The following procedure would be used after decryption to recover the original plaintext blocks: $V \oplus (V \oplus A) = A, (V \oplus A) \oplus (V \oplus A \oplus B) = B, (V \oplus A \oplus B) \oplus (V \oplus A \oplus B \oplus C) = C$, etc.

Pre-encryption block scrambling produces several beneficial side-effects. First, identical plaintext blocks are almost always represented by differing ciphertext blocks except in the rarest of cases. Secondly, scrambling makes adversarial tampering (replications, insertions, deletions, alteration) with the message stream considerably more difficult to accomplish.

Data compression prior to encryption provides another option with the added benefit of increased bandwidth. Or, one might choose to use a 6-bit character subset which was standard before the advent of 8-bit character sets. Finally, if the subject PK system is sufficiently fast and endowed with digital signature capability, plaintext could be rendered simultaneously in encrypted and authenticated form – a form of superencryption which precludes any n-gram attack since the adversary does not have the private key necessary to create the required plaintext/ciphertext pairs.

5 Conclusions

To be secure, a PK cryptosystem must satisfy several design and functional criteria with perhaps none more critical than resistance to semi-exhaustive n-gram attack – an attack whose feasibility for increasingly larger block sizes is each day made easier by an ever-growing parallelism in computer architectures [5] coupled with seemingly limitless increases in storage density and declining cost per bit of mass storage.

While any key is presumed possible for a private key system subjected to exhaustive key enumeration, only a fraction of all possible input values need be considered in an n-gram attack against vulnerable

PK systems unless special steps are taken to make its block size substantially larger than 64 bits or to increase the entropy of the input plaintext prior to encryption as noted above.

It is surely a fortunate accident of fate that the large block size of most implementations of the popular RSA cryptosystem (initially established to preclude easy factoring) also insulates against n-gram attack. However, the bad news is that nowhere have the powers of massive parallel assault been more clearly demonstrated than in the recent factoring of a 155 decimal digit number [6].

Acknowledgement

The author is grateful to former colleague and mathematician the late Jack H. Allen for permission to use his WORDS! software and dictionary in the example (as well as his computation of required prefixes, suffixes, and infixes) and for many interesting discussions on the statistics and idiosyncrasies of natural language plaintext.

References

[1] Cipra, B. 1990. Big Number Breadown. *Science*. Vol. 248: 1608, 29.

[2] Denning, D. 1982. *Cryptography and Data Security*. Reading MA: Addison Wesley.

[3] Denning, P. and W. Tichy. 1990. Highly Parallel Computation. *Science*, 1217–1222.

[4] Deavours, C. 1977. Unicity Points in Cryptanalysis. em Cryptologia 1,1 46–48.

[5] Diffie, W. and M. Hellman. 1977. Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *Computer* 10, 6, 74–84.

[6] National Bureau of Standards. 1977. Data Encryption Standard. FIPS Publication 45 , Washington DC.

[7] Rivest, R., A. Shamir and L. Adelman. 1978. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM,* Vol. 21. pp.120–126.

[8] Shannon, C. 1949. Communication Theory of Secrecy Systems. *Bell System Technical Journal* 28 656–715.

[9] Vernam, G. 1926. Cipher Printing Telegraph Systems. *Transactions AIEE*. 45: 295–301.

Biographical Sketch

Bill Wilson is a member of ACM and an early-retiree of the original Sperry Univac half of the Unisys Corporation. During his twenty-three years there he served as a consultant in the Federal Systems Division specializing in information storage and retrieval, database design technologies, and computer security.

He currently serves as an independent consultant in these areas and is co-inventor of the WARLOCK matrix-based public key crypto-system electronically published on the Internet in June 1993.