THE ENTROPY OF ENGLISH USING PPM-BASED MODELS

W. J. Teahan, John G. Cleary*

Department of Computer Science, University of Waikato, New Zealand

Over 45 years ago Claude E. Shannon estimated the entropy of English to be about 1 bit per character [16]. He did this by having human subjects guess samples of text, letter by letter. From the number of guesses made by each subject he estimated upper and lower bounds of 1.3 and 0.6 bits per character (bpc) for the entropy of English. Shannon's methodology was not improved upon until 1978 when Cover and King [6] used a gambling approach to estimate the upper bound to be 1.25 bpc from the same text. In the cryptographic community n-gram analysis suggests 1.5 bpc as the asymptotic limit for 26-letter English (Tilbourg [19]).

On the surface there is a considerable gap between these estimates and the best computer based models. For the PPM scheme [5, 12, 18] a result of around 2.4 bpc is often quoted based on Thomas Hardy's *Far from the Madding Crowd* [1, 5]. These PPM results were obtained using initially empty context models. Similar results are achievable using a block-sort algorithm [4] which has also been shown to be context based [5]. In another experiment, Brown et al. [3] used 583 million words of training text and a trigram based word model to obtain 1.75 bpc for the million word Brown Corpus [9].

The purpose of this paper is to show that the difference between the best machine models and human models is smaller than might be indicated by these results. This follows from a number of observations: firstly, the original human experiments used only 27 character English (letters plus space) against full 128 character ASCII text for most computer experiments; secondly, using large amounts of priming text substantially improves PPM's performance; and thirdly, the PPM algorithm can be modified to perform better for English text. The result of this is machine performance down to 1.46 bpc.

The importance of this goes beyond the incremental improvement in the size of compressed text. Having a computer model that achieves close to a human's is critical in areas such as speech recognition, spell-checking, OCR and language identification. It is also well-known in cryptography that removing redundancy is important prior to encryption to prevent statistical attacks [20]. It is important here that there are no models (human or otherwise) which are significantly better than the model used to remove the redundancy.

There is no reason that machine models cannot do better than humans. After all, machines can keep more consistent and accurate statistics. On the other hand, a human has access to semantic and contextual information that current computer models have no hope of accessing. This makes us pessimistic that dramatic improvements are possible in purely character-based models such as PPM.

This paper is organized as follows. The first section discusses the problem of estimating the entropy of English. The next two sections demonstrate the importance of training text for PPM and show also that its performance can be improved by "adjusting" the alphabet used. Results based on these improvements are then given, with compression down to 1.46 bpc.

*email {wjt, jcleary}@waikato.ac.nz

1068-0314/96$5.00 © 1996 IEEE

1 Estimating the entropy of English

In 1948, Shannon defined the entropy of a language [15] which he used to estimate the entropy of printed English in a paper 3 years later. Shannon's method of estimating the entropy was to have human subjects guess upcoming characters based on the immediately preceding text. He chose 100 random samples taken from Dumas Malone's *Jefferson the Virginian* [11]. Based on the number of incorrect guesses, Shannon derived upper and lower bound estimates of 1.3 bpc and 0.6 bpc.

Jelinek [10, page 476] highlights problems with Shannon's methodology. "His results would be valid only if the text they were based on was representative and large enough for his statistical estimates to converge. Representativity of English can never be established, and the text passage Shannon chose was much too short."

Cover and King [6] noted that Shannon's guessing procedure only gave partial information about the probabilities for the upcoming symbol. A correct guess only tells us which symbol a subject believes is the most probable, and not how much more probable it is than other symbols. They developed a gambling approach where each subject gambled a proportion of their current capital on the next symbol. Using a weighted average over all the subjects' betting schemes they were able to derive an upper bound of 1.25 bpc. The performance of individual subjects ranged from 1.29 bpc to 1.90 bpc. Cover and King's paper also contains an extensive bibliography on related research.

Cover and King discuss the meaning of the phrase "the entropy of English". "It should be realized that English is generated by many sources, and each source has its own characteristic entropy. The operational meaning of entropy is clear. It is the minimum expected number of bits/symbol necessary for the characterization of the text." They also point out that although the true entropy is strictly less than the upper bound, the lower bound is really a "lower bound to an upper bound" and is therefore of limited meaning.

Both Shannon's and Cover and King's approaches were based on human subjects guessing the text. Brown et al.'s approach, on the other hand, was entirely computer-based. They constructed a word trigram language model from 583 million words of training text, and then used that to estimate the entropy on the 1 million word Brown Corpus. Their approach differed from previous work, in that they were using a much larger sample of text whereas previous estimates were based on samples of at most a few hundred letters. Also, rather than use 27 character English they decided to predict all 96 printable ASCII characters.

PPM models are based upon context models using a varying number of prior characters. The models record the frequency of characters that have followed each of the contexts. For example, a particular context may be the letters "thei". All the characters that have followed this context are counted. The next time the letters "thei" occur in the text, the counts are used to estimate the probability for the upcoming character. The PPM technique blends together the context models for the varying lengths (such as "hei" and "ei") to arrive at an overall probability distribution.

A series of carefully crafted improvements have been made to the original implementation, PPMC [12], culminating in an 8% improvement for the latest implementation [18]. These implementations use a fixed upper bound to the length of the context models. Another approach is to use unbounded length context models [5]. Burrows and Wheeler's block-sorting algorithm [4] also uses unbounded contexts but unlike the PPM methods is non-adaptive. Experiments [8, 18] based on the Calgary Corpus [1] show that for current implementations performance of these approaches

are similar. However, for English text fixed length PPM models of order 5 perform up to 5% better than the others.

These improvements raise the question of how much the PPM models can be improved and whether the gap to the human entropy estimates can be closed. The first step towards this is to note that the human estimates were done using 27 character English (letters plus space) whereas the machine models are for full ASCII text. A simple experiment of converting text to this format (by replacing all sequences of non-letters with a single space, and converting all letters to lower case) show a 10% improvement in compression. Also, have we fallen into the trap of comparing apples with oranges? The apple in this case is Dumas Malone's *Jefferson the Virginian*, the text used in Shannon's experiments, and the oranges being the Brown Corpus (used in Brown et al.'s experiments) and Thomas Hardy's book (file *book1* in the Calgary Corpus). Indeed, the latter contains numerous typographical errors which have been removed in the latest electronic text available from Project Gutenberg. This corrected version improves compression by 3% to 2.24 bpc. In comparison, we have found that the entropy for the text Shannon used is in fact a lot lower, requiring just 2.01 bpc to compress the entire volume.

These musings led us to investigate ways in which PPM's performance could be improved for English text. Two such methods are described in the next two sections.

2 The use of training text to improve PPM models

One of the drawbacks with PPM is that it performs relatively poorly at the start. This is because it has not yet built up the counts for the higher order context models, so must resort to lower order models. To overcome this, a simple expedient is to use training text to prime PPM.

This raises the question of which and how much training text to use. Obviously, finding training text that is related to the text being compressed is important, preferably by the same author. Boggess et al. [2] highlight some of the problems: "we have noted that the body of work of a single writer differs significantly both from other writers and from norms for English text derived from large, multiple-source corpora."

One approach would be to find the greatest amount of text written by the same author, preferably of the same style, and about the same subject, and use this for training text. The complete works of Jane Austin is now available in the public domain: the six novels *Emma*, *Mansfield Park*, *Northanger Abbey*, *Persuasion*, *Pride and Prejudice* and *Sense and Sensibility*. Their total combined size is over 4 million characters or nearly 720,000 words. Experiments with the novel *Emma* show that compressing the last chapter using the remaining chapters plus the other five novels as training text improves compression by nearly 47% from 2.93 bpc down to 1.56 bpc. This can be further improved down to 1.48 bpc using the methods described in the next section.

However, there is a danger in all this. This text is not the same as that used by Shannon. Shannon himself found that the entropy was greater for different texts such as newspaper writing, scientific work and poetry. Indeed, any comparison made between Brown et al.'s result, which is based on the Brown Corpus, and Shannon's results is invalid because of the different source texts involved. Still, the results with *Emma* were very promising, and gave us enough encouragement to try to repeat them with Shannon's source *Jefferson the Virginian*. This text is the first volume in a series of six titled *Jefferson and His Time* which was awarded the Pulitzer Prize for History in 1975. The first volume, *Jefferson the Virginian*, was published in 1948, three years

vol.	pub. date	no. of pages	no. of words	no. of chars.	title
1	1948	423	154035	905790	*Jefferson the Virginian*
2	1951	488	182633	1081029	*Jefferson and the Rights of Man*
3	1962	506	186852	1114406	*Jefferson and the Ordeal of Liberty*
4	1970	485	175441	1058016	*Jefferson the President First Term 1801-1805*
5	1974	668	235622	1418505	*Jefferson the President Second Term 1805-1809*
6	1977	499	174940	1034505	*The Sage of Monticello*
Total		3069	1109523	6612253	*Jefferson and His Time*

Table 1: Dumas Malone's epic work *Jefferson and His Time*

before Shannon published his paper. Fortuitously (at least for our experiments), Dumas Malone then proceeded to write and publish the remaining five volumes over a period of 26 years.

In order to gain access to these one million words of text, all six volumes were scanned into the computer. Numerous errors made by the OCR software were corrected, and all footnotes, headings, page numbers and end-of-line hyphens were removed. Table 1 summarizes information about the resulting text. We estimate that the percentage of incorrect words remaining is about 0.2% based upon the number of errors found in the last chapter of the first volume. Our experiments with this text shows that over 30% improvement in compression is possible using the other five volumes as training text for the first.

The question of how much training text should be used is an important one. Brown et al. in their experiments with trigram models used a training corpus containing over 580 million words. In comparison, the training text we use for our method is just 1.1 million words for *Jefferson the Virginian* or 0.7 million words for *Emma*, although admittedly, this training text is far better tuned to the text being modelled. Experiments using training text from different authors show a marked reduction in performance, and a substantial increase in the size of the training text would be required to overcome this. This is shown in Figure 1 which illustrates how the compression of the last chapters of *Jefferson the Virginian* and *Emma* improves for different sized training texts taken from different sources, in this case the Brown Corpus, and the remaining text from *Jefferson and His Time* and the complete works of Jane Austin.

Clearly, related text performs as a better training vehicle although it seems that large amounts of unrelated text can do as well. All the curves show a quantitatively similar shape with a steep reduction between 10^4 and 10^6 characters and rather flatter beyond this. It is unclear whether this is a general result and what performance might be expected for substantially larger training texts, for example, the corpus used by Brown et al.

These results pose two related questions: "how many words does it take to 'train' a human model?" and "how many words has a human encountered in their lifetime?" Siskind [17] estimates from language learning research that children hear between $3,200$ and $50,400$ words per day. Assuming a constant rate, then a 30 year-old listener will have heard between 35 and 550 million words in their lifetime, values beyond those used in our experiments but comparable to the corpus used by Brown

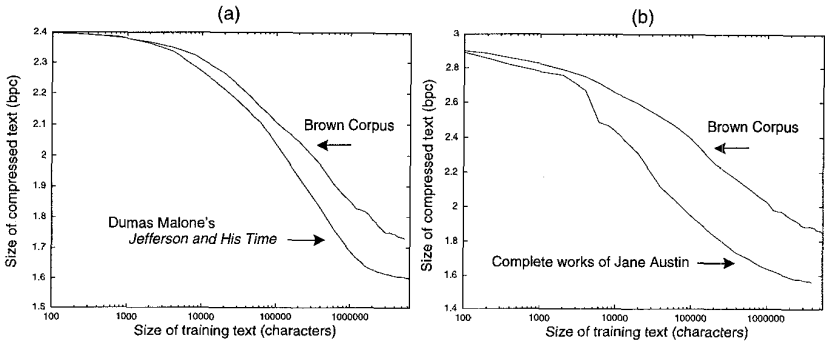

Figure 1: How training text improves compression for PPM
(a) for the last chapter of Dumas Malone's *Jefferson the Virginian*
(b) for the last chapter of Jane Austin's *Emma*

et al. It is unclear to what extent the difference between spoken and written English would affect human performance.

3 Improving the PPM model by enlarging the alphabet

In English some bigrams (pairs of characters) occur more frequently than individual letters. This led us to investigate the effect of replacing such frequent bigrams by new unique symbols. For example, one of the most frequent bigrams in English is *th*. Performance of the PPM encoder improves by up to 1% if all occurrences of *th* in the text are replaced with a single uniquely identifiable character. Curiously, a look at the history of English spelling [14] helps us understand why. In Old English, *th* was written as two single letters – the thorn and the eth, corresponding to the unvoiced and voiced sounds in *thing* and *the*. Caxton, the first English printer, sometimes used thorn's nearest typographical equivalent *Y* as there was no equivalent letter in the continental type used then by the printing presses. This may still be seen today in old-style signs such as *Ye Olde Gifte Shoppe*.

To encode text using this idea it is scanned from left to right, with the upcoming bigram replaced by a unique symbol if it occurs in a lookup table. Various encoding methods using this approach are described below, and are illustrated in Figure 2. The encoding methods fall into three categories: bigram, digram and vowel/consonant methods. It is worth noting that these methods, when combined with PPM usually perform better than PPM by itself.

Bigram based methods

Bigram coding. Bigram coding replaces frequently occurring pairs of characters with a single unique character. This presents two immediate problems: which bigrams are the most frequent, and how many bigrams should be replaced? Both problems can be solved by using frequency counts gleaned from a large corpus of representative English text (e.g. the Brown Corpus) to select the N most frequent bigrams, with N being adjusted to find the maximum compression (typically when $N \approx 104$).

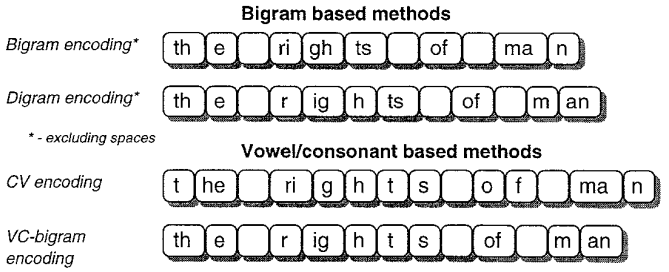

Figure 2: Various encodings for the string *"the rights of man"*

th	he	in	er	an	re	on	at	en	nd
ed	or	es	te	ti	it	is	to	st	ar
of	ng	ha	ou	al	as	nt	hi	le	se
ve	me	ne	co	ea	de	ro	ri	io	ll
ic	li	ra	be	ce	ch	ma	om	el	ho
si	wa	ca	la	ur	ta	fo	ly	no	pe
il	us	ut	di	ns	ad	ac	et	ot	ee
rs	un	ec	wh	wi	so	pr	lo	we	tr
ss	ge	nc	ct	sh	ow	ol	ai	em	ie
mo	ul	id	rt	im	ni	po	ir	ld	mi
ts	pa	na	ke	am	oo	os	gh	su	ig

Table 2: The 110 most frequent non-space bigrams from the Brown Corpus

Experimental results show that replacing bigrams containing spaces degrades performance. This is similar to the effect that removing spaces from English text has on compressed output size – the compressed output actually increases even though as much as 15% of the text may have been removed! Consequently, the bigram encoding method replaces only the most frequent bigrams that do not contain spaces. For example, the three most frequent non-space bigrams are *th*, *he* and *in*. Table 2 is the list of bigrams we used in our experiments (sorted in decreasing order of frequency from left to right).

Figure 2 illustrates how the string *the rights of man* is bigram encoded, reducing from 17 characters down to 11 symbols. Each box represents a separate symbol in the expanded alphabet, with the blank box representing the space symbol. Note that the bigram *he* in the word *the* is not replaced, despite it being the second most frequently occurring non-space bigram in the Brown Corpus. This is because the bigram *th* just before it is replaced first.

Digram coding. A *digram* (or digraph) is defined by Webster's dictionary as "a group of two successive letters whose phonetic value is a single sound (as *ea* in *bread* or *ng* in *sing*)". *Digram coding* [13] attempts to reduce the text by replacing digrams defined using the following simple algorithm. Digrams are designated as consisting of two characters – a master character (*space, n, t* and the vowels) followed by a combining

character (*space* and all letters except *j, k, q, x, y* and *z*). Replacement of digrams proceeds from left to right with spaces ignored in the same manner as bigram coding.

VOWEL/CONSONANT BASED METHODS

Another bigram-based approach uses vowels and consonants. This attempts to define bigrams which have distinct phonetic sounds (like the digram-based approach). Denes and Pinson [7] define a syllable as usually consisting of a "vowel surrounded by one or more consonants." Consequently, these methods also capture some of the regularity between syllables in the text.

CV encoding. This method of encoding replaces consonant–vowel bigrams in the text. Processing from left to right, whenever a consonant is followed by the vowel, the pair of characters is replaced by a single uniquely identifiable character, otherwise the character is left unchanged. There are 105 possible pairings (21 consonants by 5 vowels).

VC encoding. This method is similar to CV encoding except that vowel–consonant bigrams are replaced rather than the other way round.

VC–bigram encoding. Supplementing either CV or VC encoding with bigram coding presents further possibilities. After the bigrams have been replaced, a second pass through the text replaces some of the remaining bigrams. Experimental results show, however, that this approach is only effective after VC–encoding and then only for a few bigrams. The greatest improvement occurs for the bigram *th*. Most other bigrams degrade the performance. Some exceptions are the bigrams *ly*, *ph* and *oe*, although the improvement for the latter two is very slight. One could envisage searching all possible bigrams to find the combination of bigrams and VC encoding that leads to the best performed model for English text. This has not been investigated because it is computationally intractable and the resultant gains probably minor.

4 AN ESTIMATE OF THE ENTROPY OF ENGLISH

The performance of the bigram encoding methods on the last chapter of Dumas Malone's *Jefferson the Virginian* is compared with PPM on the unencoded text in Table 3. The table lists the encoding method, and the compression ratios for both trained and untrained PPM (using fixed models of order 5). The percentage improvement of each method over unencoded PPM is shown in the last column. The text was pre-converted to 27 character English for these experiments, and consists of 7985 words and 46137 characters. The best result is for the bigram encoding method at 1.488 bpc.

The results in the table graphically illustrate the importance of using training text to pre-load PPM's context models, with each method improving by over 30%. The training text used in this case was the remaining 5 volumes of Dumas Malone's work, plus all but the last chapter of the first volume, containing 1.1 million words or 6.4 million characters.

It is interesting that, although bigram encoding performs the best of all the encoding techniques, when used with untrained text it actually makes the compression 0.5% worse. To further test the robustness of bigram encoding a number of other 27 character texts were bigram encoded and compressed using untrained PPM. Table 4 shows that in all cases there was an improvement over unencoded text of 2% to 7%. The poor result in Table 3 may be due to the short length of the test text used there.

encoding method	untrained PPM (bpc)	trained PPM (bpc)	improvement (%)
unencoded text	2.402	1.598	0.00
bigram encoding	2.415	1.488	6.84
digram encoding	2.391	1.498	6.26
CV encoding	2.399	1.506	5.71
VC–bigram encoding	2.356	1.503	5.93

Table 3: Compression ratios for the last chapter of *Jefferson the Virginian*

source text	size of text (chars)	PPM (bpc)	bigram encoding (bpc)	improvement (%)
Dumas Malone's *Jefferson and His Time*	5063237	1.620	1.544	4.66
Complete works of Jane Austin	3872447	1.659	1.603	3.36
Brown Corpus	5391575	1.938	1.895	2.24
King James Bible	4139727	1.574	1.464	7.04

Table 4: Compression ratios for various texts using untrained PPM

Shannon for his experiments used 100 samples each containing 100 characters taken at random from the text. However his paper does not indicate which passages were chosen so we cannot make a direct comparison with this text. Cover and King instead chose a passage of 1490 characters (260 words) to predict the next 75 characters. Using trained PPM with bigram encoding on this short piece of text results in an estimate of 1.726 bpc. This compares with 1.488 bpc for our larger sample of 46137 characters taken from the last chapter.

This result improves even more if further related training text is available. For example, the collection of writings by Thomas Jefferson shown in Table 5 is now available in the public domain. The addition of this collection to the training text improves the estimate down to 1.461 bpc. In comparison, unrelated text does not do as well, with the addition of the entire Brown Corpus (almost doubling the size of the training text) only improving the estimate down to 1.482 bpc.

5 Summary and conclusions

A number of English texts have been compressed by machine models, with results that can be reasonably compared with previous estimates of the "entropy of English" made by Shannon and Cover and King using human subjects. We have achieved 1.46 bpc on text that Shannon estimated to have an entropy between 0.6 and 1.3 bpc. We achieved 1.726 bpc for text on which individual subjects in Cover and King's experiments obtained between 1.29 and 1.90 bpc and on which an aggregate "committee" model obtained 1.25 bpc.

These values are significantly less than the usual range of 2.3 bpc and over quoted for machine models. This contrast is accounted for by a number of factors. The human experiments were done on 27 character English, so for purposes of comparison, the same has been done for the machine models. As well, the machine models were improved by using (in order of decreasing importance):

title	no. of words	no. of chars.
A Summary View of the Rights of British America	7012	41177
Addresses, Messages, and Replies	20874	126942
Autobiography	40902	239219
First Inaugural Address	1724	10332
Indian Addresses	5847	31559
Letters	310445	1785007
Miscellany	50043	290524
Notes on the State of Virginia	66050	404735
Public Papers	58409	348392
Second Inaugural Address	2164	13034
Total	563470	3290921

Table 5: A collection of writings by Thomas Jefferson

- large ($> 10^6$ characters) amounts of English training text
- closely related training text
- replacement of frequent bigrams by single symbols
- texts with fewer typographical errors.

These results show machine models can perform in the range achieved by humans. This is important for encryption, which requires that there be no models (human or otherwise) which are significantly better than the model used to remove the redundancy.

The results also indicate that it may be difficult to obtain large improvements in machine models. On the other hand, they also indicate that the blending mechanism of PPM encoders is deficient because the best models are obtained by fixing the order of the PPM models to 5 and replacing bigrams by unique symbols. If the blending algorithms were working correctly then these steps should not be necessary.

REFERENCES

[1] T.C. Bell, J.G. Cleary, and I.H. Witten. *Text compression.* Prentice Hall, NJ, 1990.

[2] L. Boggess, R. Agarwal, and R. Davis. Disambiguation of prepositional phrases in automatically labelled technical text. *AAAI'91*, pages 155–159, 1991.

[3] P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, J.C. Lai, and R.L. Mercer. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1):31–40, 1992.

[4] M. Burrows and D.J. Wheeler. A block-sorting lossless data compression algorithm. Technical report, Digital Equipment Corporation, Palo Alto, California, 1994.

[5] J.G. Cleary, W.J. Teahan, and I.H. Witten. Unbounded length contexts for PPM. In J.A. Storer and M. Cohn, editors, *Proc. DCC'95.* IEEE Computer Society Press, 1995.

[6] T.M. Cover and R.C. King. A convergent gambling estimate of the entropy of English. *IEEE Transactions on Information Theory*, 24(4):413–421, 1950.

[7] P.B. Denes and E.N. Pinson. *The speech chain – the physics and biology of spoken language.* W.H.Freeman and Co., New York, 1993.

[8] P. Fenwick. Improvements to the block sorting text compression algorithm. Technical Report 120, University of Auckland, Auckland, New Zealand, 1995.

[9] W. Francis and H. Kucera. *Frequency analysis of English usage.* Houghton Mifflin, Boston, 1982.

[10] F. Jelinek. Self-organized language modeling for speech recognition. In A. Waibel and K. Lee, editors, *Readings in speech recognition*, pages 450–506. Morgan Kaufmann Publishers, Inc., 1990.

[11] D. Malone. *Jefferson the Virginian.* Little Brown and Co., Boston, 1948.

[12] A. Moffat. Implementing the PPM data compression scheme. *IEEE Transactions on Communications*, 38(11):1917–1921, 1990.

[13] G. Salton. *Automatic text processing.* Addison-Wesley Pub. Co., Reading, Mass., 1988.

[14] D.G. Scragg. *A history of English spelling.* Manchester University Press, Oxford Road, Manchester, 1974.

[15] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.

[16] C.E. Shannon. Prediction and entropy of printed English. *Bell System Technical Journal*, pages 50–64, 1951.

[17] J.M. Siskind. Private communication, 1995.

[18] W.J. Teahan. Probability estimation for PPM. In *Proc. of the N.Z. Comp. Sci. Research Students' Conf., 1995.* Univ. of Waikato, Hamilton, New Zealand, 1995.

[19] H.C.A. Tilbourg. *An introduction to cryptology.* Kluwer Academic Publishers, 1988.

[20] W.J. Wilson. Chinks in the armor of public key cryptosystems. Technical Report 94/3, University of Waikato, Hamilton, New Zealand, 1994.