MIT/LCS/TM-256

ON THE NUMBER OF CLOSE-AND-EQUAL PAIRS OF BITS IN A STRING (WITH IMPLICATIONS ON THE SECURITY OF RSA'S L.S.B.)

Oded Goldreich

March 1984

On the Number of Close-and-Equal Pairs of Bits in a String (with Implications on the Security of RSA's L.S.B)

Oded Goldreich
Laboratory for Computer Science
MIT,room NE43-836,Cambridge,MA 02139

Abstract

We consider the following problem: Let s be a n-bit string with m ones and $n-m$ zeros. Denote by $CE_t(s)$ the number of pairs, of equal bits which are within distance t apart, in the string s. What is the minimum value of $CE_t(\cdot)$, when the minimum is taken over all n-bit strings which consists of m ones and $n-m$ zeros?

We prove a (reasonably) tight lower bound for this combinatorial problem.

Implications, on the cryptographic security of the least significant bit of a message encrypted by the RSA scheme, follow. E.g. under the assumption that the RSA is unbreakable; there exist no probabilistic polynomial-time algorithm which guesses the least significant bit of a message (correctly) with probability at least **0.725** , when given the encryption of the message using the RSA. **This is the best result known concerning the security of RSA's least significant bit.**

Keywords: Cryptography, Combinatorial Analysis, the RSA Scheme, Bit Security, Combinatorial Bounds, Bit-String Properties, Public Key Cryptosystems.

Supported by a Weizmann Postdoctoral Fellowship

1. Introduction

This paper combines a combinatorial study with the application of its results to the analysis of a cryptological question. (The combinatorial problem is fully defined and solved in Sec. 2.)

1.1. Cryptological Background

The importance of the notion of "partial information" to cryptographic research has gained wide recognition through the pioneering works of Blum and Micali [BM] and Goldwasser and Micali [GM]. In this paper we consider a much more specific question: the cryptographical security of the least significant bit of a message encrypted by the RSA scheme (hereafter referred to as RSA's l.s.b) .

The RSA encryption scheme was presented by Rivest, Shamir and Adleman [RSA]. It is the best known implementation of the notion of a Public Key Cryptosystem, which was suggested by Diffie and Hellman [DH]. Encryption using the RSA is done by raizing the message to a known exponent, e, and reducing the result modulo a known composite number, N, the factorization[1] of which is kept secret. The inverse of e in the multiplicative group $Z^*_{\varphi(N)}$ is used for decryption and is kept secret. It is widely believed that the RSA is hard to break. This means that an adversary who does not know the **secret** (e^{-1} *mod* $\varphi(N)$) will not be able to compute the message from its encryption (i.e. to invert the encryption function).

However, even under this unbreakability assumption; it might be the case that the RSA leaks some "valuable" partial information. I.e. it might be that given the ciphertext, one can compute some function of **half** of the bits of the plaintext. Proving that, under the unbreakability assumption, this is infeasible will make the RSA much more attractive. This seems to be a high tool. Research attempts are meanwhile focused at the feasibility of guessing correctly the least significant bit of the plaintext (i.e. RSA's l.s.b.)[2].

By saying that *RSA's l.s.b is p-secure* we mean that guessing it correctly with probability at least p is as hard as inverting the RSA. Consider an **oracle** that when given the encryption (using the RSA) of a message guesses the least significant bit of the message correctly with probability p. Such an oracle will be called a *p-oracle for RSA's l.s.b* . Clearly, the existence of a polynomial time algorithm that inverts the RSA using a p-oracle for RSA's l.s.b implies that RSA's l.s.b is p-secure.

It is believed that RSA's l.s.b is $(\frac{1}{2}+\epsilon)$-secure , for arbitrary small constant ϵ. Proving this statement might be a major breakthrough on the way to proving that any "valuable" partial information about the message encrypted by the RSA is as hard to get as inverting the RSA. Progress towards this goal has been slow but consistant, in the recent years.

---

[1] To be exact, N is the produce of two large primes, p and q. $\varphi(\cdot)$ is the Euler's totient function, thus $\varphi(pq) = (p-1)(q-1)$.

[2] Nevertheless, results have been acheived also w.r.t. other kinds of partial information. For details consult [BCS] and [VV2].

The first step was taken by Goldwasser Micali and Tong [GMT] who proved that RSA's l.s.b is $(1 - \frac{1}{|N|})$-secure, where $|N|$ is the size of the RSA's modulus.

Ben-Or, Chor and Shamir greatly improved this result by proving that RSA's l.s.b is $(\frac{3}{4} + \epsilon)$-secure, where ϵ is fixed and arbitrary small. Their paper [BCS] contains an algorithm which inverts the RSA function. Their algorithm uses a $(\frac{3}{4} + \epsilon)$-oracle for RSA's l.s.b (in order) to **determine** the parities of certain multiples of the original message. For further details consult [BCS] or [VV2].

Vazirani and Vazirani [VV1] have presented a very sophisticated modification of the algorithmic procedure used by Ben-Or, Chor and Shamir. The theme of their modification is a much better use of the oracle answers. They showed that their modification is guaranteed to succeed when given access to a 0.741-oracle for RSA's l.s.b. Recently, they have improved their analysis by showing that their modification is guaranteed to succeed even if it uses a 0.732-oracle.

Using the combinatorial results obtained in this paper, we show that the Vazirani and Vazirani algorithm is guaranteed to succeed when it uses a 0.725-oracle for RSA's l.s.b. Other observations w.r.t the Vazirani and Vazirani algorithm as well as w.r.t other inverting algorithms are also implied.

1.2. Our Results

The following problem occured to us when trying to improve Ben-Or, Chor and Shamir's result [BCS]:

> Let s be a n-bit string with m ones and $n - m$ zeros. Two bits in the string s are said to be t-close if they are within distance t apart. Denote by $CE_t(s)$ the **number of pairs of equal t-close bits in the string s** . What is the minimum value of $CE_t(\cdot)$, over all n-bit strings which consists of m ones and $n - m$ zeros?

In Sec.2 we prove a (reasonably) tight lower bound on this combinatorial problem. With respect to proving the "amount" of security of the least significant bit of the RSA, this is a double-edged-sword:

(1) It provides **a powerful tool** for analyzing certain algorithms for inverting the RSA using an $(\frac{1}{2} + \delta)$-oracle for RSA's l.s.b .

For example **the algorithm proposed by Vazirani and Vazirani [VV1] is shown to work when it uses any 0.725-oracle for RSA's l.s.b** (i.e. δ=0.225). This establishes the best result known conserning the security of RSA's l.s.b .

(2) It **points out the weakness** of various proof techniques for determining the cryptographic security of RSA's l.s.b .

For example the Vazirani and Vazirani algorithm [VV1] may fail to invert if it uses a $\frac{2}{3}$-oracle for RSA's l.s.b .

These implications will be discussed in Sec. 3 . We believe that the combinatorial result has also other implications.

2. The Combinatorial Results

In this section we give a formal definition of the combinatorial problem, discussed in the introduction, and prove a (reasonably) tight lower bound on it.

2.1. Definitions

Let $s = (s_0, s_1, s_2, .., s_{|s|-1})$ be a binary string of length $|s|$. We denote by $sh_i(s)$ the string which result from s by the application of i left cyclic shifts. I.e:

$$sh_i(s) = (s_i, s_{i+1}, s_{i+2}, .., s_{i+|s|-1}) ,$$

where indices are considered modulo $|s|$.

Define the *i-overlap* of a string, s, to be the number of positions which agree in s and $sh_i(s)$. The i-overlap of s will be denoted by $over_i(s)$, i.e.

$$over_i(s) \;=\; Hamming(s \equiv sh_i(s)) ,$$

where $\equiv$ denotes the bit by bit equal operation and $Hamming(s)$ denotes the number of ones in s . Note that $over_i(s) = |\{j: 0 \leq j < |s| \wedge s_j = s_{j+i}\}|$.

Denote by AverOver(s,t) the average over the i-overlaps of s for $i \in \{1, 2, .., t\}$. I.e.

$$AverOver(s,t) = \frac{1}{t}\sum_{i=1}^{t} over_i(s)$$

We remind the reader that $CE_t(s)$ was used to denote the number of pairs, of equal bits which are within distance t apart, in the string s . I.e.

$$CE_t(s) = |\{(i,j): 0 \leq i < j < n \wedge s_i = s_j \wedge j - i \leq t\}| ,$$

where $n = |\ s\ |$.

Clearly, $CE_t(s)=\sum_{i=1}^{t}|\{j: 0 \leq j < n \wedge s_j = s_{j+i}\}|$. Thus,

$$CE_t(s) \;=\; t \cdot \text{AverOver}(s,t) .$$

When evaluating $CE_t(s)$ consider "lines" which connect **equal t-close bits in s** (i.e. positions that contain equal values and are less than t bits apart in the string s). These lines are hereafter called *overlines*. Note that $CE_t(s)$ is nothing but the number of overlines in the string s.

Let n and m be integers such that $0.5n \leq m < n$. Let $\delta = \frac{m-0.5n}{n}$. We denote by S_n^δ the set of n-bit binary strings with $m=(0.5+\delta)n$ ones (and $n-m$ zeros).

Denote by Aver(n,δ,t) the minimum value of AverOver($\cdot$,t) divided by n, when minimized over all strings in S_n^δ. I.e.

$$\text{Aver}(n,\delta,t) = min_{s \in S_n^\delta} \{ \tfrac{1}{n} \cdot \text{AverOver}(s,t) \}.$$

It is straightforward to see that for every $s \in S_n^\delta$, AverOver(s,n)$=(0.5+2\delta^2)n$.

In this section we study Aver(n,δ,t) for arbitrary t, $t<n$. We obtain non-trivial results, as the surprising fact that Aver(n,0,t) converges to $\sqrt{2}-1 \approx 0.414$, when $\frac{n}{t}$ and t are large enough.

2.2. Propositions

We will assume throughout this section that $t \leq \frac{1}{2}(n-2)$. We will analyze Aver(n,δ,t) as follows: first we will show that the minimum of $CE_t(\cdot)$ is acheived by strings which belong to a restricted subset of S_n^δ; and next we will minimize $CE_t(\cdot)$ over this subset. This will establish a lower bound on Aver(n,δ,t). The upper bound will be implied by the proof of the lower bound, since this proof specifies a string $s \in S_n^\delta$ for which $CE_t(s) \approx nt \cdot$Aver(n,δ,t).

2.2.1. Reduction into a restricted subset

In this subsection we will show that when analysing Aver(n,δ,t) it is enough to consider strings in S_n^δ which have the following property:

The string contains **no** "short 3-alternations substring". A *short 3-alternations substring* is a substring of the form $\sigma\tau^+\sigma^+\tau$ and length less than $t+2$, where $\sigma \neq \tau \in \{0,1\}$. (Here, and throughout this paper, σ^+ denotes a non-empty string of σ's.)

Proposition 1: $over_i(s) = over_i(sh_j(s))$

Prop. 1 follows directly from the definitions which consider strings as if they were cycles. From this point on, we also take the liberty of doing so.

Proposition 2: Let $\sigma_j \in \{0,1\}$, for $1 \leq j \leq 2t$. Let α be a binary string. Let $n_{\tau_1\tau_2} = CE_t(\sigma_1\sigma_2\cdots\sigma_t\tau_1\tau_2\sigma_{t+1}\sigma_{t+2}\cdots\sigma_{2t}\alpha)$. Then $n_{10} - n_{01} = 2(\sigma_1 - \sigma_{2t})$.

proof: Note that the difference between $n_{\tau_1\tau_2}$ and $n_{\tau_2\tau_1}$ is only due to the existence or non-existence of overlines between σ_1 and τ_1 and between τ_2 and σ_{2n} . Details are left to the reader.

Qed

Note that *switching* τ_1 and τ_2 in the string $\sigma_1\sigma_2\cdots\sigma_t\tau_1\tau_2\sigma_{t+1}\sigma_{t+2}\cdots\sigma_{2t}\alpha$ results in the string $\sigma_1\sigma_2\cdots\sigma_t\tau_2\tau_1\sigma_{t+1}\sigma_{t+2}\cdots\sigma_{2t}\alpha$. The latter string has more overlines (than the former one) only if $\sigma_1 = \tau_2 \neq \tau_1 = \sigma_{2n}$. Note that the latter string has less overlines if $\sigma_1 = \tau_1 \neq \tau_2 = \sigma_{2n}$.

Proposition 3: Let α be a binary string and let x, y, z, u be integers such that $x + y \geq t$ but $y + z < t$. Then:

(i) $CE_t(\sigma\tau^x\sigma^y\tau^{z-1}\sigma\tau\alpha) \leq CE_t(\sigma\tau^x\sigma^y\tau^z\sigma\alpha)$.

(ii) $CE_t(\sigma\tau^x\sigma^y\tau^{z-1}\sigma\tau\sigma^{u-1}\tau^{t-u}\sigma\alpha) < CE_t(\sigma\tau^x\sigma^y\tau^z\sigma^u\tau^{t-u}\sigma\alpha)$.

(iii) $CE_t(\sigma\tau^x\sigma^y\sigma\tau^z\alpha) \leq CE_t(\sigma\tau^x\sigma^y\tau^z\sigma\alpha)$.

proof:

Part (i) follows by switching in $\sigma\tau^x\sigma^y\tau^z\sigma\alpha$ the σ on the l.h.s. of α with the τ on the l.h.s. of that σ; and recalling Prop. 2. (Notice that the symbol in $\sigma\tau^x\sigma^y\tau^z\sigma\alpha$ which is t bits to the left of "the switched τ" is also a τ.)

Part (ii) follows by switching in $\sigma\tau^x\sigma^y\tau^z\sigma^u\tau^{t-u}\sigma\alpha$ the σ on the l.h.s. of $\sigma^{u-1}\tau^{t-u}\sigma\alpha$ with the τ on the l.h.s. of that σ; and recalling again Prop. 2. (Notice that the symbol in $\sigma\tau^x\sigma^y\tau^z\sigma^u\tau^{t-u}\sigma\alpha$ which is t bits to the right of "the switched σ" is also a σ.)

Part (iii) follows by z sequential applications of part (i).

Qed

Proposition 4: Let $s \in S_n^\delta$ be a binary string such that $CE_t(s) = n \cdot t \cdot \mathrm{Aver}(n,\delta,t)$. (I.e. s is a string with minimum number of overlines among all strings in S_n^δ .) Then there exist a string, $s' \in S_n^\delta$, such that :

(i) The string s' contains **a** substring of the form 10^+1^+0 the length of which is **at least** $t+2$.[3]

(ii) $CE_t(s') < CE_t(s) + t^2$.

proof: Note first that s is not of the form 0^+1^+. (Otherwise switching the adjacent 0 and 1 in s, results in a string with less overlines.)

Consider an arbitrary substring, α, of length t in s. Let z denote the number of zeros in α ($t-z$ is the number of ones in α).

Case 1: If $z = 0$ or $z = t$ then the proposition follows, when $s' = s$.

Case 2 ($0 < z < t$): Let σ_L and σ_R be the bits adjacent to α in the string s. Replacing $\sigma_L\alpha\sigma_R$ by $\sigma_L 0^z 1^{t-z}\sigma_R$ in the string s results in a string s'. Note that the number of overlines within $\sigma_L\alpha\sigma_R$ is equal to the number of overlines within $\sigma_L 0^z 1^{t-z}\sigma_R$. Also note that the number of overlines between the 0^z1^{t-z}-block and the rest of s' (excluding σ_L and σ_R) is at most $t(t-1)$. Thus, $CE_t(s') \leq CE_t(s) + t(t-1)$ and the proposition follows.

Qed

Proposition 5: Let $s' \in S_n^\delta$ be a string, with **minimum** number of overlines, which satisfies Prop. 4 . Then with no loss of generality, the string s' contains **no** substring of the form 10^+1^+0 the length of which is **less than** $t+2$. Furthermore, the string s' contains **at most one** substring of the form 01^+0^+1 the length of which is **less than** $t+2$.

We remind the reader that $CE_t(s') < nt\mathrm{Aver}(n,\delta,t)+t^2$ and that $s' \in S_n^\delta$.

proof: By the hypothesis, s' contains a substring of length at least $t+2$ which has the form 10^+1^+0. The following is a sketch of the proof:

Starting at such a substring and scanning s' cyclicly (from left to right) we apply switches to make sure that all **scanned** substrings of either the form 10^+1^+0 or the form 01^+0^+1 are of length at least $t+2$. We stop before scanning the last unscanned 01^+0^+1 substring. Noticing that the above process does not increase the number of overlines, we are done.

The proof proceeds as follows:

By Prop. $4_{(i)}$, we can assume, w.l.o.g, that $s' = 10^i1^j0\alpha$, where $i+j \geq t$ and $\alpha \in \{0,1\}^*$. We define the following *scanning procedure* and apply it to $s_{scan} = 1\$\$0^i1^j\$0\alpha$. (\$\$ denotes the "starting position" and \$ denotes the "current position" in the scanning.)

[3] We remind the reader that σ^+ denotes a non-empty string of σs.

procedure scanning($\sigma\gamma_0\tau^x\sigma^y\$\gamma_1\tau^{z-z'}\gamma_2\tau^{z'}\sigma\beta_1\gamma_3\beta_2$) *recursive*;
  [$\sigma,\tau\in\{0,1\}$, $\sigma\neq\tau$, $\gamma_0,\gamma_1,\gamma_2,\gamma_3\in\{\lambda,\$\$\}$ and $\beta_1,\beta_2\in\{0,1\}^*$.]
if $\gamma_1=\$\$$ *then return*($\sigma\tau^x\sigma^y\tau^z\sigma\beta_1\beta_2$); [terminates.]
[$\gamma_1=\lambda$] *if* $y+z\geq t$ *then* [considers next block.]
  return(scanning($\tau\sigma^y\tau^z\$\gamma_2\sigma\beta_1\gamma_3\beta_2\sigma\gamma_0\tau^{x-1}$));
[$\gamma_1=\lambda$ and $y+z<t$] [transfers one σ.]
  return(scanning($\sigma\gamma_0\tau^x\sigma^y\sigma\$\gamma_1\tau^{z-z'}\gamma_2\tau^{z'}\beta_1\gamma_3\beta_2$));
end;

It is possible to verify that the string scanning(s_{scan}) satisfies the statement of the proposition. For details, consult the Appendix (Sec. 6.1).

Qed

Proposition 6: Let $s'\in S_n^\delta$ be a string as in Prop. 5. Then there exist a string $s''\in S_n^\delta$ such that:

(i) The string s'' contains no substring of the form 10^+1^+0 the length of which is less than $t+2$.

(ii) The string s'' contains no substring of the form 01^+0^+1 the length of which is less than $t+2$.

(iii) $CE_t(s'')<CE_t(s')+t^2$.

proof: By the hypothesis s' has no 10^+1^+0 substring and at most one 01^+0^+1 substring of length less than $t+2$. Assume that such a unique 01^y0^z1 substring of length less than $t+2$ exists; i.e. $y+z<t$. Replace this substring in s' by the substring 00^z1^y1 resulting in a string s''. Note that s'' satisfies both (i) and (ii). To conclude note that $CE_t(s'')<CE_t(s')+t^2-t$. [The number of overlines within 01^y0^z1 is equal to the number of overlines within 00^z1^y1; the number of overlines between the 0^z1^y-block and the rest of s'' is lest than $t(t-1)$.] The proposition follows.

Qed

We remind the reader that our objective is to given a good lower bound on Aver(n,δ,t)=$\min_{s\in S_n^\delta}\frac{1}{nt}CE_t(s)$. Note that we have restricted our attention to strings that donot have short 3-alternations substrings; i.e. substrings of the form 01^+0^+1 or 10^+1^+0 which have length less than $t+2$. This is sufficient since there exist such a string, namely s'', that has approximately the minimun number of overlines. I.e. $CE_t(s'')<nt\text{Aver}(n,\delta,t)+2t^2$. Formally we define R_n^δ to be the set of strings which belong to S_n^δ and do not have short 3-alternating substrings. $\text{Aver}_R(n,\delta,t)$ will denote $\min_{r\in R_n^\delta}\frac{1}{nt}CE_t(r)$. Clearly,

Proposition 7: $\text{Aver}(n,\delta,t)\leq\text{Aver}_R(n,\delta,t)<\text{Aver}(n,\delta,t)+\frac{2t}{n}$.

proof: By Prop. 4,5 and 6, $s''\in R_n^\delta$ and $nt\text{Aver}_R(n,\delta,t)\leq CE_t(s'')<CE_t(s)+2t^2=nt\text{Aver}(n,\delta,t)+2t^2$. Thus, the proposition follows.

Qed

Let us define even a more restricted subset of S_n^δ: The set MR_n^δ is the subset of strings which belong to R_n^δ and do not have *long homogenous substrings*; i.e. substring of

the form σ^{t+1}, where $\sigma \in \{0,1\}$. Also, $\mathrm{Aver}_{MR}(n,\delta,t)$ will denote $\min_{r \in MR_n^\delta} \frac{1}{nt} CE_t(r)$. Let us first give a tight lower bound on $\mathrm{Aver}_{MR}(n,\delta,t)$ and only later prove that this bound is approximately also a bound for $\mathrm{Aver}_R(n,\delta,t)$.

2.2.2. Lower bound for $\mathrm{Aver}_{MR}(n,\delta,t)$

Recall that each of the strings in $MR_n^\delta \subseteq S_n^\delta$ has the following properties:

(i) The string contains no short 3-alternating substrings.

(ii) The string contains no long homogenous substrings.

We will relay on the above properties of the strings in MR_n^δ in order to bound $\mathrm{Aver}_{MR}(n,\delta,t)$. Given a string $r \in MR_n^\delta$ we will introduce an expression, for $CE_t(r)$, which depends only on the numbers of bits in each maximal substrings of consecutive equal bits. In other words, we will introduce a localized counting of $CE_t(r)$.

Definition: We say that b is a block (an all-σ block) of the string r if it is a maximal substring of equal bits. I.e. $b = \sigma^+$ and $r = \tau b \tau \alpha$, where $\tau \neq \sigma$ and α is an arbitrary string.

Denotations: Let q denote the number of all-zero [all-one] blocks in r. Beginning from an arbitrary position between an all-one block and an all-zero block and going cyclically from left to right; number the blocks of consecutive zeros [ones] by 0,1,2,...,$(q-1)$. Denote by z_i the number of zeros in the i-th all-zero-block and by y_i the number of ones in the i-th all-one-block. I.e., $r = 0^{z_0}1^{y_0}0^{z_1}1^{y_1}0^{z_2}1^{y_2}\cdots 0^{z_{q-1}}1^{y_{q-1}}$.

Proposition 8: Overlines occur (in r) only **either** within a block **or** between two consecutive blocks (of the same bit).

proof: Consider any substring of the form $10^+1^+0^+1$ in r. By Prop. 6, the length of this substring exceeds $t+1$ and therefore no overlines exist between the extreem 1's. Similiar observation holds for any $01^+0^+1^+0$ substring. Thus, the proposition follows.

Qed

Remark: Note that Prop. 8 holds even if $r \in R_n^\delta$.

This suggests to evaluate the number of overlines (in r) by counting the "contribution" of each (homogeneous) block to it. This counting is hereafter referred as the *Block-Localized Counting (BLC)* and proceeds as follows:

Block-Localized Counting (with respect to a block of length l in r):

(i) The number of overlines **within the block**, denoted I_l.

(ii) The number of overlines **between** bits of the **blocks** neighbouring this block (i.e the first block on its left and the first block on its right), denoted B_l.

Note that I_l and B_l are easy to evaluate and can be used to express $CE_t(r)$. Namely,

Proposition 9:

(i) $CE_t(r) = \sum_{i=0}^{q-1}((I_{y_i} + B_{y_i}) + (I_{z_i} + B_{z_i}))$, where $r = 0^{z_0}1^{y_0}0^{z_1}1^{y_1}\cdots 0^{z_{q-1}}1^{y_{q-1}}$.

(ii) For $l < t$, $I_l = \binom{l}{2}$ and $B_l = \sum_{i=1}^{t-l} i$.

(iii) For $l = t$, $I_l = \binom{t}{2}$ and $B_l = 0$.

proof: Part (i) follows by observing that each overline is counted exactly once.

To evaluate B_l consider, w.l.o.g, the substring $00^i1^l0^k1$. If $i+l<t$ then the number of overlines between the leftmost 0 and the 0's to the right of 1^l-block is $t-(l+i)$. This is due to the fact that (by $r \in MR_n^\delta$) $l+k \geq t$. Also note that if $i+l \geq t$ then there are no overlines between the leftmost 0 and the 0's to the right of the 1^l-block. Thus, $B_l = \sum_{i=0}^{t-l-1}(t-l-i)$ if $l<t$; and $B_l = 0$ otherwise.

Clearly, for $l \leq t+1$, $I_l = \binom{l}{2}$. Thus, the proposition follows.

Qed

Remark: Note that for $l>t$, $I_t = \binom{t}{2} + (l-t)t$ and $B_t = 0$. (Note that for $k>0$, $CE_t(\sigma^{t+k}) = CE_t(\sigma^{t+k-1}) + t = CE_t(\sigma^t) + kt$.) However such substrings donot exist in a string which belongs to MR_n^δ.

Evaluating $I_l + B_l$ we get

Proposition 10: The contribution (to the BLC) of one l-bit long block (in r) is:

$$f(l) = l^2 - (t+1)l + \tfrac{t^2+t}{2} .$$

proof: Recall that $r \in MR_n^\delta$ and thus $l \leq t$. using Prop. $9_{(ii)}$ and $9_{(iii)}$, we get $f(l) = \binom{l}{2} + \frac{1}{2}(t-l)(t-l+1)$ and the proposition follows.

Qed

Note that the contribution of all the all-zero blocks to the number of overlines (in r) only depends on the way the zeros are partitioned among the all-zero blocks. (I.e. it is independent of the way the ones are partitioned among the all-one blocks.) This contribution amounts to:

$$g(z_0, z_1, .., z_{q-1}) = \sum_{i=0}^{q-1} f(z_i) ,$$

where $r = 0^{z_0}1^{y_0}0^{z_1}1^{y_1} \cdots 0^{z_{q-1}}1^{y_{q-1}}$.

Note that $g(\cdot,\cdot,\cdots,\cdot)$ is a quadratic form and therefore

Proposition 11: For fixed q, t and k, the minimum value of the function $g(x_0, x_1, .., x_{q-1})$ subject to the constraint $k = \sum_{i=0}^{q-1} x_i$, is obtained at $x_0 = x_1 = \cdots = x_{q-1} = \frac{k}{q}$.

proof: Note that $g(x_0, x_1, .., x_{q-1}) = \sum_{i=0}^{q-1}(x_i^2 - (t+1)x_i + \frac{t(t+1)}{2}) = \sum_{i=0}^{q-1} x_i^2 - (t+1)\cdot k + \frac{1}{2}t(t+1)\cdot q$. Since $\sum_{i=0}^{q-1} x_i^2$ subject to $k = \sum_{i=0}^{q-1} x_i$ is minimum when the x_is are equal, the proposition follows.

Qed

Thus, the minimum number of overlines is achieved if all the all-zero-blocks [all-one-blocks] are of the same size. This yields

Proposition 12: Let $Q = \{q \in Integers : \frac{m}{t} \leq q \leq n-m\}$. Then:

$$ntAver_{MR}(n,\delta,t) \geq \min_{q\in Q}\{q \cdot (f(\tfrac{m}{q}) + f(\tfrac{n-m}{q}))\} .$$

We remind the reader that $m = (0.5+\delta)n$.

proof: Note that the number of blocks in a string, $r \in MR_n^\delta$, must be in Q. The proposition follows immediately from the definitions of $\mathrm{Aver}_{MR}(n,\delta,t)$ the functions $f(\cdot)$ and $g(\cdot,\cdot,\cdots,\cdot)$, and Prop. $9_{(i)}$,10 and 11.

Qed

Elaborating the r.h.s. expression of Prop. 12 we get

Proposition 13: $\mathrm{Aver}_{MR}(n,\delta,t) \geq \min_{q\in Q}\{h_n^\delta(q)\}$, where

$$h_n^\delta(q) = \frac{t+1}{n}\cdot q + \frac{(0.5+2\delta^2)n}{t}\cdot\frac{1}{q} - \frac{t+1}{t} .$$

proof: $\frac{q}{nt}(f(\frac{m}{q}) + f(\frac{n-m}{q}))$

$= \frac{q}{nt}((\frac{m}{q})^2 - (t+1)\frac{m}{q} + \frac{t(t+1)}{2} + (\frac{n-m}{q})^2 - (t+1)\frac{n-m}{q} + \frac{t(t+1)}{2})$

$= \frac{q}{nt}(t(t+1) + \frac{n^2-2mn+2m^2}{q^2} - \frac{(t+1)n}{q}) = \frac{t+1}{n}q + \frac{(0.5+2\delta^2)n}{tq} - \frac{t+1}{t} = h_n^\delta(q)$.

Qed

Note that

Proposition 14: The minimum of the function $h_n^\delta(\cdot)$ is obtained at:

$$q_{min} = \sqrt{\frac{0.5+2\delta^2}{t(t+1)}}\cdot n ;$$

and the minimum value, $h_n^\delta(q_{min})$, is:

$$v_t^\delta = \sqrt{(2+8\delta^2)\cdot\frac{t+1}{t}} - \frac{t+1}{t} .$$

Thus, $\mathrm{Aver}_{MR}(n,\delta,t) \geq v_t^\delta$. All that is left is to derive a lower bound for $\mathrm{Aver}_R(n,\delta,t)$.

2.2.3. Lower bound for $\mathrm{Aver}_R(n,\delta,t)$ and $\mathrm{Aver}(n,\delta,t)$

In this subsection we show that a string, $r_0 \in R_n^\delta$, with minimum overlines can be transformed into a string $r_0' \in MR_{n'}^{\delta'}$, such that $n' \approx n$, $\delta' \approx \delta$ and $CE_t(r_0') \approx CE_t(r_0)$. We conclude by using this fact and the lower bound for $\mathrm{Aver}_{MR}(n,\delta,t)$, to introduce a lower bound for $\mathrm{Aver}_R(n,\delta,t)$.

Proposition 15: Let $r_0 \in R_n^\delta$ be a string with minimum number of overlines; i.e. $CE_t(r_0) = nt\mathrm{Aver}_R(n,\delta,t)$. Then:

(i) For $\sigma \in \{0,1\}$, either r_0 contains no substring of more than t consecutive σ's or r_0 contains no block of less than t consecutive σ's. Futhermore, w.l.o.g, r_0 contains atmost one substring of more than t consecutive σ's.

(ii) If $t > \frac{\frac{1}{2}+\delta}{\frac{1}{2}-\delta}$ then r_0 has no substring of the form σ^{2t}.

(iii) If $t \leq \frac{\frac{1}{2}+\delta}{\frac{1}{2}-\delta}$ then $\mathrm{Aver}(n,\delta,t)=2\delta$.

(iv) If $t > \frac{\frac{1}{2}+\delta}{\frac{1}{2}-\delta}$ then there exist a $k < t$, a $\delta' \geq \delta$ and a $r_0' \in MR_{n+k}^{\delta'}$ such that $CE_t(r_0) \geq CE_t(r_0') - kt$.

proof:

Part (i): Note that omitting one σ from a substring that contains more than t σ's decreases the number of overlines by exactly t. Adding one σ to a block of k σ's increases the number of overlines by t if $k \geq t$, and by less than t if $k < t$. Thus, w.l.o.g, r_0 contains at most one substring of more than t σ's. Also, note that r_0 can not contain both a substring of more than t σ's and a block of less than t σ's. (Otherwise omiting one σ from the first substring and adding it to the second one, will result in a new string which is also in R_n^δ but has less overlines than the string r_0. This is in contradiction to the hypothesis.) Thus, part (i) of the proposition follows.

Part (ii): Assume on the contrary that r_0 contains a σ^{2t} substring.

Case 1 ($\sigma = 0$): Since the number of 1's is at least as much as the number of 0's, r_0 contains a 11 substring. Omitting one of the ones in the 11 substring and inserting it in the middle of the 0^t0^t substring decreases the number of overlines, in contradiction to the hypothesis.

Case 2 ($\sigma = 1$): By part (i) above and since $t > \frac{\frac{1}{2}+\delta}{\frac{1}{2}-\delta}$, r_0 contains a 00 substring. Contradiction follows as in Case 1.

Part (iii): Note that the number of overlines in a string $s \in S_n^\delta$ is at least $nt - 2(\frac{1}{2}-\delta)nt = 2\delta nt$. On the other hand, $CE_t((1^t0)^i 1^j) = nt - 2it$, where $n = i(t+1)+j$. Note that if $t \leq \frac{\frac{1}{2}+\delta}{\frac{1}{2}-\delta}$ then such a string exists. Part (iii) of the proposition follows.

Part (iv): By part (i), if r_0 contains a 0^{t+k} substring then it contains also a 1^{t+k} substring. Also r_0 contain at most one 0^t0^+ $[1^t1^+]$ block. Thus, w.l.o.g, we consider the longest 1^+ substring. Let l denote its length. By part (ii) it is enough to consider two cases:

Case 1 ($l \leq t$): Let $r'_0 = r_0$, $k = 0$ and $\delta' = \delta$. By the above $r'_0 \in MR_n^\delta$.

Case 2 ($t < l < 2t$): Note that r_0 contains a 00 substring. Let $k = 2t - l$ and r'_0 be the string which results from r_0 by the following procedure:

```
add k ones to the longest 1+ block (yielding a 1^2t block);
if r0 contains a 0^(t+u) block  (when u > 0)
then do begin
    omit u zeros from the 0^(t+u) block;
    insert them in the middle of the 1^2t block; end
else do begin
    omit 1 zero from a 00 substring;
    insert it in the middle of the 1^2t block; end
```

Let $\delta' = \delta + \frac{(0.5-\delta)k}{n+k}$. Note that $\delta' = \frac{(0.5+\delta)n+k-0.5(n+k)}{n+k}$. Also note that by the above, $r'_0 \in MR_{n+k}^{\delta'}$ and $CE_t(r'_0) < CE_t(r_0) + kt$.

Thus, part (iv) of the proposition follows.

Qed

We conclude by using Prop. $15_{(iv)}$ and the lower bound for $\mathrm{Aver}_{MR}(n,\delta,t)$, to introduce lower bounds for $\mathrm{Aver}_R(n,\delta,t)$ and $\mathrm{Aver}(n,\delta,t)$.

Proposition 16: If $t > \frac{\frac{1}{2}+\delta}{\frac{1}{2}-\delta}$ then

(i) There exist $0 \leq k < t$ and $\delta' \geq \delta$ such that $\mathrm{Aver}_R(n,\delta,t) > \mathrm{Aver}_{MR}(n+k,\delta',t) - \frac{t}{n}$.

(ii) $\mathrm{Aver}_R(n,\delta,t) > v_t^\delta - \frac{t}{n}$.

(iii) $\mathrm{Aver}(n,\delta,t) > v_t^\delta - \frac{3t}{n}$.

proof:

By Prop. $15_{(iv)}$, $\mathrm{Aver}_R(n,\delta,t) = \frac{1}{nt} CE_t(r_0) > \frac{1}{nt}(CE_t(r_0') - t^2) \geq \mathrm{Aver}_{MR}(n+k,\delta',t) - \frac{t}{n}$. Thus, part (i) of the proposition follows.

By Prop. 13 and 14, $\mathrm{Aver}_{MR}(n+k,\delta',t) \geq v_t^{\delta'} \geq v_t^\delta$. Thus, part (ii) follows.

Combining the above with Prop. 7, part (iii) follows.

Qed

2.3. The Main Results

Throughout this section we assume that $\frac{\frac{1}{2}+\delta}{\frac{1}{2}-\delta} < t \leq \frac{1}{2}(n-2)$.

Lower Bound Lemma: $\mathrm{Aver}(n,\delta,t)$ is at least

$$(\sqrt{(2+8\delta^2)\cdot\tfrac{t+1}{t}} - \tfrac{t+1}{t}) - \tfrac{3t}{n} .$$

proof: The Lemma follows immediately from Prop. 14 and $16_{(iii)}$.

Qed

Upper Bound Lemma: $\mathrm{Aver}(n,\delta,t)$ is at most

$$(\sqrt{(2+8\delta^2)\cdot\tfrac{t+1}{t}} - \tfrac{t+1}{t}) + \tfrac{t+1}{n} + \tfrac{1}{2t^2} .$$

proof: The Lemma follows from observing that the proof of the lower bound specifies the structure of a string which achieves minimum $CE_t(\cdot)$ among all strings in MR_n^δ. The only problem in constructing such a string is that non-integer numbers, of blocks and block sizes, may appear. However, we will show that the overlap added by the round-up of the number of blocks is less than $\frac{t+1}{n}$; while the overline added by the round-up of the blocks' sizes is less than $\frac{1}{2t^2}$.

Let q_{min} denote, as in Prop. 14, the value on which $h_n^\delta(\cdot)$ is minimized and let $\nu = \lceil q_{min} \rceil - q_{min}$. Note that $h_n^\delta(\lceil q_{min} \rceil) - h_n^\delta(q_{min}) = \nu\frac{t+1}{n} + \frac{(0.5+2\delta^2)n}{t}(\frac{1}{\lceil q_{min} \rceil} - \frac{1}{q_{min}})$. Thus, $h_n^\delta(\lceil q_{min} \rceil) < h_n^\delta(q_{min}) + \frac{t+1}{n}$. [A better bound can be obtained if the number of blocks is rounded up to $\lfloor q_{min} \rfloor$. One can prove that $h_n^\delta(\lfloor q_{min} \rfloor) < h_n^\delta(q_{min}) + O((\frac{t+1}{n})^2)$.]

Consider the partition of the zeros among the q zero-blocks. Let $z = \frac{n-m}{q}$ and assume $\nu_0 = z - \lfloor z \rfloor > 0$. Consider the partition of $\lfloor z \rfloor$ zeros to each of the

first k_0 zero-blocks and $\lceil z \rceil$ to each of the rest. Note that $\frac{k_0}{q} = 1 - \nu_0$. Define $d_0 = (k_0 \cdot f(\lfloor z \rfloor) + (q - k_0) \cdot f(\lceil z \rceil)) - (q \cdot f(z))$. Using $k_0 = (1 - \nu_0)q$, $\lfloor z \rfloor = z - \nu$ and $\lceil z \rceil = z - \nu + 1$, we get $d_0 = q((1 - \nu_0)(z - \nu_0)^2 + \nu_0(z - \nu_0 + 1)^2 - z^2) = q\nu_0(1 - \nu_0) \leq q(\frac{1}{2})^2 < \frac{n}{4t}$. In case $\nu_0 = 0$, let $k_0 = q$ and note that $d_0 = 0$. The same applies to the partition of the 1's and the evaluation of d_1. Note that the above partitions define a string, $s_q \in S_n^\delta$, such that $CE_t(s_q) - nt \cdot h_n^\delta(q) = d_0 + d_1 < 2 \cdot \frac{n}{4t}$.

We coclude by noting that

$$\mathrm{Aver}(n,\delta,t) \leq \frac{1}{nt} CE_t(s_{\lfloor q_{min} \rfloor}) < \frac{1}{nt}(nt \cdot h_n^\delta(\lceil q_{min} \rceil) + \frac{n}{2t}) < h_n^\delta(q_{min}) + \frac{t+1}{n} + \frac{1}{2t^2} .$$

Qed

Evaluating the expressions in the above lemmas we get

Corollary 1:

(i) $\sqrt{2} - 1 - O(\frac{1}{t}) < \mathrm{Aver}(n,\mathbf{0},t) < \sqrt{2} - 1 + O(\frac{1}{t^2}) + O(\frac{t}{n})$.

(ii) For $t \geq 2500$ and $n > 300000 \cdot t$, $\mathrm{Aver}(n,\mathbf{0.177},t) > \frac{1}{2} + 0.0001$.

(iii) For $t \geq 500$ and $n > 10000 \cdot t$, $\mathrm{Aver}(n,\mathbf{0.225},t) > \mathbf{0.55} + 0.0001$.

(iv) For every $2500 < t < \frac{n}{10000}$ and $\delta \leq 0.176$, $\mathrm{Aver}(n,\delta,t) < \frac{1}{2}$.

(v) For every $500 < t < \frac{n}{10000}$ and $\delta \leq 0.224$, $\mathrm{Aver}(n,\delta,t) < 1 - 2\delta$.

2.4. Additional Definitions and Results

In this section we define a different, yet related, combinatorial problem. Instead of considering the average overlap over all "small"[4] shifts; we consider the maximum overlap obtained by one of the "small" shifts.

Let us define an *i-overline* to be a line which connects a pair of equal bits which are (exactly) at distance i apart.

Denote by $\mathrm{MaxOver}(s,t)$ the maximum over the i-overlaps of s for $i \in \{1, 2, .., t\}$. I.e.

$$\mathrm{MaxOver}(s,t) = max_{1 \leq i \leq t} \{ over_i(s) \} .$$

Denote by $\mathrm{Max}(n,\delta,t)$ the minimum value of $\mathrm{MaxOver}(s,t)$ divided by n, when minimized over all strings in S_n^δ . I.e.

$$\mathrm{Max}(n,\delta,t) = min_{s \in S_n^\delta} \{ \frac{1}{n} \cdot \mathrm{MaxOver}(s,t) \}.$$

Clearly,

Proposition 17: $\mathrm{Max}(n,\delta,t) \geq \mathrm{Aver}(n,\delta,t)$.

This establishes a trivial lower bound on $\mathrm{Max}(n,\delta,t)$. We donot beleive that this bound is tight; however we failed to prove a better one. On the other hand the following proposition yields an upper bound on $\mathrm{Max}(n,0,t)$.

Proposition 18: ((i) is folklore and (ii) appears in van Lint[L])

(i) For every De-Bruijn Sequence[5], s, of length 2^k and every i, $i \in \{1, 2, .., k-1\}$

[4] Here, "small" means not greater than t.

[5] The 2^k-bit long string $(s_0, s_1, s_2, ..., s_{2^k-1})$ is a De-Bruijn Sequence if (when considered in circular order) it contain as substrings all possible bit-strings of length k.

$$over_i(s) = \tfrac{1}{2} \cdot 2^k \ .$$

(ii) For every k there exists a Shortened De-Bruijn Sequence[6], s, of length $2^k - 1$ such that for every i, $i \in \{1, 2, .., 2^k - 2\}$,

$$over_i(s) = 2^{k-1} - 1 \approx \tfrac{1}{2} \cdot (2^k - 1) \ .$$

Using Prop. 18 we also obtain an upper bound on Max(n,δ,t); i.e.

Proposition 19: [Here q is an integer.]

(i) For $t+1 = l = 2^k - 1$, $n = ql$ and $\delta = \frac{l+q-1}{2n}$, Max($n$,$\delta$,$t$) $\leq \frac{1}{2} + \delta - \frac{1}{t+1} + \frac{1}{n}$.

(ii) Max(n,δ,t)$\leq$Max(n,δ,$t+1$).

(iii) Max(n,δ,t)$< \frac{1}{2} + \delta + O(\frac{t}{n})$.

proof: Part (ii) follows easily from the definition of MaxOver.

Let s be a Shortened De-Bruijn Sequence as in Prop. $18_{(ii)}$ (i.e. $over_i(s) = 2^{k-1} - 1$, when $0 < i < 2^k - 1$). The proof of parts (i) and (iii) consists of constracting strings which are shown to have "low" MaxOver. (These MaxOvers will set an upper bound on the coresponding Max($\cdot$, $\cdot$, $\cdot$).) The constractions use the string s^+ as a substring, where $k \approx \log_2 t$. Additional 1's are used, to outnumber the zeros in the constracted strings, in case $\delta > 0$. Details can be found in the Appendix (Sec. 6.2).

Qed

2.5. Historical Remark

The combinatorial results presented in Sec. 2.2 and Sec. 2.3 as well as Corollary 4 (of Sec. 3.2) were obtained during September 1983.

The exact statement of the VV-Theorem was communicated to the author on November 21st; the results presented in Sec. 2.4 and Sec. 3.2 were obtained during the rest of November 1983.

[6] A Shortened De-Bruijn Sequence, of length $2^k - 1$, is a 2^k-long De-Bruijn Sequence in which a zero has been omitted from the all-zero block of length k .

3. On the Cryptographic Security of the RSA's L.S.B

In this section we apply the results of the privious section to the analysis of algorithms which invert the RSA encryption function when given access to an oracle for the least significant bit of the encrypted message. This implies results (concerning the security of RSA's l.s.b.) which fall into the following three categories:

(i) A 0.725-security result (for RSA's l.s.b)

(ii) Conditional improvements of the above result. I.e. results which will hold if some conjecture is proven.

(iii) Bounds on the possibility of improvements using current techniques.

3.1. Specific Background

Our 0.725-security result is based on Vazirani and Vazirani work [VV1], which is an improvement of Ben-Or Chor and Shamir [BCS] work. In this subsection we sketch some of the ideas used in these nice works.

3.1.1. A Sketch of Ben-Or Chor and Shamir Algorithmic Procedure

The essence of the Inverting Algorithm:

The plaintext is reconstructed , from its encryption, by running a g.c.d procedure on two multiples[7] of it. The values of these multiples (as well as the values of all multiples discussed hereafter) are "small"[8]. A Modified Binary G.C.D algorithm is used. To operate, this algorithm needs to know the parity of multiples of the plaintext. Thus, it is provided with a *subroutine* that determines the parity of these multiples.(see [BCS])

Determining Parity using an Oracle which may err:

The *subroutine* determines the parity of a multiple ,kx, of the plaintext ,x, by using an $(\frac{1}{2}+\delta)$-oracle for RSA's l.s.b as follows. It picks a random r and asks the oracle for the parity (i.e. l.s.b) of both rx and $rx+kx$ feeding it in turn with $E(rx) = E(r)E(x)$ and $E((r+k)x) = E(r+k)E(x)$[9] . The oracle's answers are processed according to the following observation. Since kx is "small" with very high probability $rx < rx+kx$. Then, the parity of kx is equel to 0 if the parities of rx and $rx+kx$ are identical; and equal to 1 otherwise. This is repeated many times; every repetition (instance) is called a kx-measurement (or a toss of the kx-coin). Note that the outcome of a kx-measurement is correct if the oracle was correct on both rx and $rx+kx$. The outcome is correct also if the oracle was wrong on both queries (but this fact is not used in [BCS]).

---

[7] All integers and operations are considered modulo ,N, the RSA's modulus.

[8] Here and throughout the rest of the paper "small" means bounded by a very small fraction of the RSA's modulus.

[9] $E(M)$ denotes the RSA encryption function. Recall that $E(M) = M^e \ (mod\ N)$, where N and e are respectively the RSA's modulus and exponent.

(Trivial) Measurement Analysis:

A kx-coin toss is correct with probability at least 2δ .

(This suffices if $\delta = \frac{1}{4} + \epsilon$, see [BCS])

3.1.2. A Sketch of Vazirani and Vazirani Modification of the BCS-Procedure

Distinguishing a Good Coin from a Bad one:

For $\delta < \frac{1}{4}$; if when running a Monte-Carlo experiment on a kx-coin toss, more than a 1-2δ fraction of the answers agree on some value, then this is the correct value.(In such a case the coin is said to be *distinguishably good.* See [VV1])

Using Distinguishably Good Coins:

Let t be a fixed constant and K be a set of cardinality $O(\log N)$. If *for every* $k \in K$ *there exist a* $1 \leq j \leq t$ *such that the* $(j \cdot kx)$*-coin is distinguishably good* then one can determine the parity of kx. (This is done by replacing every kx-measurement, of the subroutine, by a set of $O(\log \log N)$ measurements, see [VV1]). (The above condition will be referred to as the *Distinguishability Condition.*)

Vazirani and Vazirani combined the above sketched ideas to an algorithm that inverts the RSA using a $(\frac{1}{2} + \delta)$-oracle. It remained to be shown that when given certain oracles for RSA's l.s.b the Distinguishability Condition holds. In [VV1] Vazirani and Vazirani proved that the Distinguishability Condition holds for any 0.741-oracle for RSA's l.s.b.; in [VV2] they improved their analysis and showed that this condition holds for any 0.732-oracle.

3.2. Cryptographic Implications of our Combinatorial Results

It is easy to show that the Distinguishability Condition is equivalent to the following condition, hereafter referred to as the *Big-Advantage Condition* : for some fixed t, Max(N,δ,t)$>1-2\delta+\epsilon$.

(Use oracle transformation through multiplication by the inverse of kx mod N. Note that if the inverse does not exist it is feasible to factor N and inverting the RSA becomes easy.) This was also observed by Vazirani and Vazirani [VV2].

Thus, we can summerize Vazirani and Vazirani's [VV1] work by the following

VV-Theorem: Let N be the RSA's modulus and t be a fixed constant. If Max(N,δ,t)$>1-2\delta+\epsilon$ then any $(\frac{1}{2} + \delta)$-oracle for RSA's l.s.b can be used to efficiently invert the RSA. (In other words: if the Big Advantage Condition holds for δ then RSA's l.s.b is $(\frac{1}{2} + \delta)$-secure.)

By our results, the Big-Advantage Condition holds for $\delta \geq 0.225$. Namely, using the VV-Theorem, Prop. 17 and Corollary $1_{(iii)}$ we get

Corollary 2: Any 0.725-oracle for the least significant bit of the RSA can be efficiantly used to invert the RSA.

In other words

Theorem: RSA's l.s.b. is 0.725-secure.

Note that **the result of corollary $1_{(iii)}$ is tight. Thus under the condition $\mathbf{Aver}(n,\delta,t) > 1 - 2\delta + \epsilon$, the result of Corollary 2 is optimal.** However, $\mathrm{Aver}(n,\delta,t) > 1 - 2\delta + \epsilon$, is more than is needed to satisfy the Big-Advantage Condition. (Recall that the Big-Advantage Condition requires only that $\mathrm{Max}(n,\delta,t) > 1 - 2\delta + \epsilon$.) Thus, any improvement of the current lower bound on $\mathrm{Max}(n,\delta,t)$ will yield an improvement of the result of Corollary 2. We beleive that $\mathrm{Max}(n,\delta,t) > \mathrm{Aver}(n,\delta,t)$ and thus that such an improvement is possible. Furthermore we conjecture that

Conjecture 1: $\mathrm{Max}(n,\delta,t) \approx \frac{1}{2} + \delta$.

Combined with the VV-Theorem this implies

Corollary 3: If Conjecture 1 is valid then RSA's l.s.b. is $(\frac{2}{3} + \epsilon)$-secure, for arbitrary small fixed ϵ.

Note that **under the Big-Advantage Condition the "result" of Corollary 3 is optimal.** This is due to Prop. $19_{(iii)}$ which states that $\mathrm{Max}(n,\delta,t) \leq \frac{1}{2} + \delta$. Thus, using the VV-Theorem (or any proof technique which requires that the Big-Advantage Condition holds) one cannot hope to prove that RSA's l.s.b is $\frac{2}{3}$-secure.

Let us conclude by pointing out that the full power of the results obtained in section 2.3 was not used; however, we conjecture that it can be used. Namely,

Conjecture 2: Let N be the RSA's modulus and $t << N$. If $\mathrm{Aver}(N,\delta,t) > \frac{1}{2} + \epsilon$ then any $(\frac{1}{2} + \delta)$-oracle for RSA's l.s.b can be used to efficiently invert the RSA. (In other words: if $\mathrm{Aver}(N,\delta,t) > \frac{1}{2} + \epsilon$ then RSA's l.s.b is $(\frac{1}{2} + \delta)$-secure.)

The condition of the statement of Conjecture 2 is hereafter referred to as the *Average-Advantage Condition.* By Corollary $1_{(ii)}$, the Average-Advantage Condition is satisfied by $\delta = 0.177$; thus

Corollary 4: If Conjecture 2 is valid then the RSA's l.s.b is 0.677-secure.

Note that $\delta = 0.177$ is the minimum for which the Average-Advantage Condition is satisfied. Thus no progress beyond the $\delta = 0.177$ point can be made through the Average-Advantage Condition; i.e. when relying on it one cannot hope to prove that RSA's l.s.b is 0.676-secure.

Note that in Corollary 4 the missing part to reach the stated result is the algorithm that will use the analysis. (The analysis of the question which oracles satisfy the Avarage-Advantage Condition is complete!) However, in the case of the Big-Advantage Condition improved results can still be achieved (just) by improving the analysis of the combinatorial problem (see Corollary 3).

4. Conclusion

We have solved a combinatorial problem and have shown how to use this solution to improve knowledge on the security of RSA's l.s.b . We have also pointed out possible directions for further improvement of our result. Improved results can be obtained by either conducting a better combinatorial analysis of $\mathrm{Max}(\cdot,\cdot,\cdot)$ or by suggesting an inverting algorithm based on the Average-Advantage Condition.

However such improvements will not suffice to show that RSA l.s.b. is $\frac{2}{3}$-secure. We believe that any improvement in the results concerning the security of RSA's l.s.b , beyond the $\frac{2}{3}$ point (which is still out of reach), must make use of additional properties of the RSA.

5. Acknowledgements

I am indebted to Tom Leighton for teaching me how to count (overlaps).

I would like to thank Ron Rivest for guiding me with his insightful suggestions.

I thank Vijay Vazirani for a private presentation of the .741 result and for his remarks.

It is my pleasure to thank Michael Ben-Or, Benny Chor, Shafi Goldwasser, Hans Heller, Silvio Micali, Gary Miller and Avi Wigderson for very helpful discussions, useful ideas and consistent encouragement.

I am most thankful to Dassi Levi for her unique existence.

6. Appendix: Details of the proofs of Prop. 5 and Prop. 19

6.1. Details of the proof of Prop. 5

Recall that s' has the minmum number of overlines among all strings which satisfy Prop. 4. Also recall that $s_{scan} = 1\$\$0^i1^j\$0\alpha$, where $i+j \geq t$ and $s' = 10^i1^j0\alpha$. The *scanning procedure* is hereafter recursively defined:

procedure scanning$(\sigma\gamma_0\tau^x\sigma^y\$\gamma_1\tau^{z-z'}\gamma_2\tau^{z'}\sigma\beta_1\gamma_3\beta_2)$ *recursive*;
  $[\sigma,\tau \in \{0,1\},\ \sigma \neq \tau,\ \gamma_0,\gamma_1,\gamma_2,\gamma_3 \in \{\lambda,\$\$\}$ and $\beta_1,\beta_2 \in \{0,1\}^*.]$
  if $\gamma_1 = \$\$$ *then* [terminate.]
(1) *return*$(\sigma\tau^x\sigma^y\tau^z\sigma\beta_1\beta_2)$;
  $[\gamma_1 = \lambda]$ *if* $y+z \geq t$ *then* [consider next block.]
(2) *return*(scanning$(\tau\sigma^y\tau^z\$\gamma_2\sigma\beta_1\gamma_3\beta_2\sigma\gamma_0\tau^{x-1})$);
  $[\gamma_1 = \lambda$ and $y+z < t]$ [transfer one σ.]
(3) *return*(scanning$(\sigma\gamma_0\tau^x\sigma^y\sigma\$\gamma_1\tau^{z-z'}\gamma_2\tau^{z'}\beta_1\gamma_3\beta_2)$);
end;

Recall that s_{scan} is the argument by which scanning is invoked in the first time. Let $s^{(i)}_{scan} = \sigma_i\gamma^{(i)}_0\tau^{x_i}_i\sigma^{y_i}_i\$\gamma^{(i)}_1\tau^{z_i}_i\gamma^{(i)}_2\sigma_i\beta^{(i)}_1\gamma^{(i)}_3\beta^{(i)}_2$ denote the argument of scanning in its i-th invokation. (Clearly, $s^{(1)}_{scan} = s_{scan}$.) It is easy to verify the following claims:

Claim 1: Exactly one of the $\gamma^{(i)}_j$'s in $s^{(i)}_{scan}$ is a non-empty word (i.e. \$\$); in case $i=1$ it is $\gamma^{(i)}_0$. The number of \$'s in $s^{(i)}_{scan}$ is exactly 3. [By induction on i.]

Denote by $s^{(i)}_{\$\$BET\$}$ the (non-empty) substring of $s^{(i)}_{scan}$, the two leftmost symbols of which are \$ signs and so is its rightmost symbol. Let $d_i = |s^{(i)}_{\$\$BET\$}|$ (d_i is defined only if scanning was invoked at least i times).

Claim 2: If d_{i+1} is defined then $d_{i+1} > d_i$. Thus, scanning terminates after at most $|s^{(1)}_{scan}|$ invokations. [Note that both commands (2) and (3) of the scanning procedur increase the distance between \$\$ and \$.]

Claim 3: $x_i + y_i \geq t$. [By induction on i.]

Denote by $s^{(i)}_{omit}$ the string which results from $s^{(i)}_{scan}$ when omitting the \$ signs which appear in it (i.e. in $s^{(i)}_{scan}$). Denote by T the number of times scanning was invoked.

Claim 4: For every $i < T$, if $y_i + z_i \geq t$ then $s^{(i+1)}_{omit} = s^{(i)}_{omit}$. [Notice that in case $y_i + z_i \geq t$, command (2) is executed.]

Claim 5: For every $i < T$, if $y_i + z_i < t$ then $CE_t(s^{(i+1)}_{omit}) \leq CE_t(s^{(i)}_{omit})$. [Notice that in case $y_i + z_i < t$, command (3) is executed. Recall Claim 3 and Prop. $3_{(iii)}$.]

Claim 6: $s^{(T)}_{\$\$BET\$} = \$\$\tau^{z_T}_T\sigma_T\beta^{(T)}_1\beta^{(T)}_2\sigma_T\tau^{x_T}_T\sigma^{y_T}_T\$$. [Consider scanning's termination condition.]

Definition: We say that $\rho_1\rho_2^x\rho_3^y\rho_4$ is a *troublesome string* if $\rho_2,\rho_3 \in \{0,1\}$, $\rho_1,\rho_4 \in \{0,1,\$\}$, $x+y<t$ and $\rho_j \neq \rho_{j+1}$, for all $1 \leq j \leq 3$.

Claim 7: $s^{(1)}_{\$\$BET\$}$ does not contain a troublesome string.

Claim 8: For every $i < T$, if $y_i + z_i \geq t$ and $s^{(i)}_{\$\$BET\$}$ does not contain a troublesome string, then $s^{(i+1)}_{\$\$BET\$}$ does not contain a troublesome string. [Notice that in case $y_i + z_i \geq t$, command (2) is executed.]

Claim 9: For every $i < T$, if $y_i + z_i < t$, $\gamma_2^{(i)} = \lambda$ and $s^{(i)}_{\$\$BET\$}$ does not contain a troublesome string, then $s^{(i+1)}_{\$\$BET\$}$ does not contain a troublesome string. [Notice that in case $y_i + z_i < t$, command (3) is executed; however, since $\gamma_2^{(i)} = \lambda$, the transferred σ_i is not in $s^{(i)}_{\$\$BET\$}$.]

Let $\beta_1^{(i)}\beta_2^{(i)} = \sigma_i^{u_i}\tau_i^{v_i}\sigma_i\beta_0^{(i)}$.

Claim 10: If $\gamma_2^{(i)} = \$\$$, $y_i + z_i < t$ and $s^{(i)}_{\$\$BET\$}$ does not contain a troublesome string, then $u_i + v_i \geq t$. [Note that if $\gamma_2^{(i)} = \$\$$ then $s^{(i)}_{scan} = \sigma_i\tau_i^{x_i}\sigma_i^{y_i}\$\tau_i^{z_i-z'_i}\$\$\tau^{z'_i}\sigma_i\sigma_i^{u_i}\tau_i^{v_i}\sigma_i\beta_0^{(i)}$. By the non-existence of a troublesome substring in $s^{(i)}_{\$\$BET\$}$, we have $1+u_i+v_i \geq t$. Note that $1+u_i+v_i = t$ leads to contradiction with our hypothesis that s' has the minimum number of overlines (recall Claim 3 and Prop. $3_{(ii)}$).]

Claim 11: For every $i < T$, if $y_i + z_i < t$, $\gamma_2^{(i)} = \$\$$ and $s^{(i)}_{\$\$BET\$}$ does not contain a troublesome string, then $s^{(i+1)}_{\$\$BET\$}$ does not contain a troublesome string. [Notice that in case $y_i + z_i < t$, command (3) is executed; however, by Claim 10 the claim holds.]

Claim 12: For every $i \leq T$, $s^{(i)}_{\$\$BET\$}$ contains no troublesome strings. [By induction on i, using claims 7, 8,9 and 11.]

Combining the above we conclude that:

(i) The string scanning(s_{scan}) is well defined. [By Claim 2.]

(ii) $CE_t(scanning(s_{scan})) = CE_t(s')$. [By Claims 4 and 5, and recalling that s' has minimum overlines.]

(iii) The string scanning(s_{scan}) contains no substring of the form 10^+1^+0 the length of which is less than $t+2$. Furthermore, it contains at most one substring of the form 01^+0^+1 the length of which is less than $t+2$. [By Claims 6 and 12.]

Thus, scanning(s_{scan}) satisfies the statement of Prop. 5.

6.2. Details of the proof of Prop. 19

Let s be a Shortened De-Bruijn Sequence as in Prop. $18_{(ii)}$ (i.e. $over_i(s) = 2^{k-1} - 1$, when $0 < i < 2^k - 1$).

Part (i): Consider the string $s' = s^{q-1}1^l$. Let n denote the length of s' and m denote its Hamming weight (i.e. number of 1's). Then $n = ql$ and $m = (q-1)2^{k-1} + 2^k - 1 = \frac{1}{2}((q+1)l + q - 1)$. Recall that $\delta = \frac{m-0.5n}{n}$. Thus, we have $\delta = \frac{l+q-1}{2ql}$. Note that $\text{MaxOver}(s^q,t) = q \cdot \frac{t}{2}$. Let us show, now, that $\text{MaxOver}(s',t) \leq \text{MaxOver}(s^q,t) + \frac{1}{2}(l+1)$.

Note that s' is the string which results from s^q when substituting one of the s substring by a 1^l substring. Consider the change in the i-overlap under this substitution.

Let s_j denote the j-th bit in s , $0 \leq j < 2^k - 1$. W.l.o.g, consider the following two cases:

case 1: $(s_j = s_{j+i})$ **subcase 1.1:** $(j + i < 2^k - 1)$ substituting s by 1^l does not change this i-overline between the j-th position and the $(j+i)$-th position. **subcase 1.2:** $(j + i > 2^k - 2)$ substituting s by 1^l can only eliminate the i-overlines between these positions and positions in the neighbouring substrings. Note that in both subcases no new i-overlines were created.

case 2: $(s_j = 0,\ s_{j+i} = 1)$ **subcase 2.1:** $(j + i < 2^k - 1)$ substituting s by 1^l creates a new i-overline between the j-th position and the $(j+i)$-th position. **subcase 2.2:** $(j + i > 2^k - 2)$ substituting s by 1^l creates a new i-overline between the $(j+i)$-th position and the j-th position in the neighbouring substring. Note that in both subcases, one i-overline was created, by the substitution, per each position j . Thus, the number of these new i-overlines is $2^{k-1} = \frac{1}{2}(l+1)$.

Thus, $\text{MaxOver}(s',l-1) \leq \text{MaxOver}(s^q,l-1) + \frac{1}{2}(l+1)$. To conclude note that $\text{Max}(n,\delta,l-1) \leq \frac{1}{n}\text{MaxOver}(s',l-1) \leq \frac{1}{n}(q\frac{l-1}{2} + \frac{1}{2}(l+1)) = \frac{1}{2} + \delta - \frac{1}{l} + \frac{1}{n}$

Part (iii): Let $k = \lceil \log_2 t + 2 \rceil$ and $l = 2^k - 1$. By part (ii) and $t \leq l-1$, $\text{Max}(n,\delta,t) \leq \text{Max}(n,\delta,l-1)$. Let $m = (\frac{1}{2} - \delta)n$ and $q = 1 + \lfloor \frac{n-m}{2^{k-1}-1} \rfloor$. Consider two cases:

Case 1: $(b = m - ((q-1)2^{k-1} + l) \geq l)$ Consider the string $s' = s^{q-1}1^l0^c1^b$, where $c = (n-m) - (q-1)(2^{k-1}-1)$. Notice that $s' \in S_n^\delta$ and that $\text{MaxOver}(s',l-1) \leq \text{MaxOver}(s^{q-1}1^l,l-1) + (b+c)$. As in part (i), we have $\text{MaxOver}(s^{q-1}1^l,l-1) \leq q\frac{l-1}{2} + \frac{1}{2}(l+1)$. Note that $n = ql + b + c$, $m = (q-1)2^{k-1} + l + b$ and $\delta = \frac{l+q-1+b-c}{2n}$. Thus, $\text{Max}(n,\delta,l-1) \leq \frac{1}{n}\text{MaxOver}(s',l-1) \leq \frac{1}{n}(q\frac{l-1}{2} + \frac{1}{2}(l+1) + b + c) = \frac{1}{2} + \delta + \frac{c-q+1}{n} = \frac{1}{2} + \delta + \nu$, where $\nu = (1 + \frac{1}{l})\frac{c}{n} + \frac{b}{ln} - \frac{1}{l}$. Note that $c < \frac{l-1}{2}$. Therefore, $\nu < \frac{(l+1)(l-1)+2b}{2ln} - \frac{1}{l} < \frac{l}{2n}$.

Case 2: $(m - ((q-1)2^{k-1} + l) < l)$ Consider the string $s' = s^{q-1}1^b0^c$, where $b = m - (q-1)2^{k-1}$ and $c = (n-m) - (q-1)(2^{k-1}-1)$. Note that $b < 2l$, $c < \frac{l-1}{2}$, $n = (q-1)l + b + c$, $m = (q-1)\frac{l+1}{2} + b$, $\delta = \frac{q-1+b-c}{2n}$ and $\text{MaxOver}(s',l-1) < (q-1)(2^{k-1}-1) + b + c + l = n(\frac{1}{2} + \delta) + c + 1 + l - q$. Thus, $\text{Max}(n,\delta,l-1) < \frac{1}{2} + \delta + \frac{3l+5}{2n} - \frac{1}{l}$.

7. References

[BCS] Ben-Or,M., Chor,B., and Shamir,A., "On the Cryptogrsphic Security of Single RSA Bits", *15th ACM Symp. on Theory of Computation*, April 1983, pp. 421-430

[BM] Blum,M., and Micali,S., "How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits", to appear in the *SIAM Jour. on Computing*

[DH] Diffie,W., and Hellman,M.E., "New Directions in Cryptography", *IEEE Trans. on Inform. Theory*, Vol. IT-22, No. 6, November 1976, pp. 644-654

[GM] Goldwasser,S., and Micali,S., "Probabilistic Encryption", to appear in the *JCSS special issue from the 14th STOC*

[GMT] Goldwasser,S., Micali,S., and Tong,P., "Why and How to Establish a Private Code on a Public Network", *Proc. of the 23rd IEEE Symp. on Foundation of Computer Science*, November 1982, pp. 134-144

[L] van Lint,J.H., *Combinatorial Theory Seminar, Eindhoven University of Technology*, Lecture Notes in Mathematics, SpringVerlag, 1974, pp. 90-91.

[RSA] Rivest,R.L., Shamir,A., and Adleman,L., "A Method for Obtaining Digital Signature and Public Key Cryptosystems", *Comm. of the ACM* ,Vol.21, February 1978, pp. 120-126

[VV1] Vazirani,U.V., and Vazirani,V.V., "RSA's l.s.b is .741 Secure", presented in *Crypto83*, August 1983.

[VV2] Vazirani,U.V., and Vazirani,V.V., "RSA Bits are .732 Secure", preprint, November 1983.