<u>Theorem 5'</u>: The SHWP problem is undecidable even if $\Sigma \triangleq \Sigma'$.

<u>Proof</u>: Modify the proof of Theorem 5 as follows: Replace $i^{\pi}_{x_\sigma}$ $[d^{\pi}_{x_\sigma}]$ by $i^{\pi}_{x_0} \cdot i^{\pi}_{x_\sigma}$ $[d^{\pi}_{x_\sigma} \cdot d^{\pi}_{x_0}]$ and $E^{\pi}_{x_\sigma}$ $[D^{\pi}_{x_\sigma}]$ by $i^{\pi}_{x_1} \cdot i^{\pi}_{x_\sigma}$ $[d^{\pi}_{x_\sigma} \cdot d^{\pi}_{x_1}]$, for $\pi \in \{L,R\}$ and $\sigma \in \{0,1\}$. Replace $i^{\pi}_{a_\sigma}$ $[d^{\pi}_{a_\sigma}]$ by $i^{\pi}_{a_0} \cdot i^{\pi}_{a_\sigma}$ $[d^{\pi}_{a_\sigma} \cdot d^{\pi}_{a_0}]$ and $E^{\pi}_{a_\sigma}$ $[D^{\pi}_{a_\sigma}]$ by $i^{\pi}_{a_1} \cdot i^{\pi}_{a_\sigma}$ $[d^{\pi}_{a_\sigma} \cdot d^{\pi}_{a_1}]$, for $\pi \in \{L,R\}$ and $\sigma \in \{0,1\}$.

Q.E.D.

## 6. ON FINDING THE SHORTEST INSECURITY STRING AND THE POWER OF NAME APPENDING

In this section we return to the definitions of Section 2.

<u>6.1</u> We consider, first, the following problem: Given an insecure p-party ping-pong protocol, find the length of a shortest insecurity string (when the length of a string is defined to be the number of operators in it).

This problem may either be of interest to a cryptanalyst who wishes to read a message transferred via an insecure protocol or for a protocol-designer who is seeking a protocol which is "practically" secure (i.e. its shortest insecurity string is infeasibly long).

Our solution is essentially the one given in DEK, for the security-testing problem, with the modification of using a priority queue, instead of an ordinary queue used there (see Appendix E). Note that the implementation of a priority queue costs a factor of log n both in the time and space complexities; thus the time complexity of the algorithm is $O(n^3 \cdot \log n)$, when n is the length of the protocol.

Obviously, a cryptanalyst would like to find the shortest insecurity string and not only to know its length. To this end another modification of the algorithm would be needed (see again Appendix E). The modification consists of leaving tracks, during the computation of the length of the shortest insecurity string, so that this string can latter be written in time linear to its length.

6.2 Next, we present a distributed problem which demonstrates the power of protocols which use a name appending mechanism (or any similar pair of operators, $\sigma$ and $\tau$, such that $\sigma\tau \equiv \lambda$ is a cancellation rule but $\tau\sigma \equiv \lambda$ is not).

Define a Multi-Reader Protocol (MRP) to be a secure multi-party ping-pong protocol such that at least two participants (in addition to the initiator) read the initial message.

Theorem 6: For $\Sigma = \{i_x, d_x, E_x, D_x : x \in N\}$ there exists a MRP.

Proof: Consider the following $(t+1)$-party protocol where $\underline{x} = (x_o, x_1, \ldots, x_t)$ is the vector of variable-users:
$\alpha_1(\underline{x}) \triangleq E_{x_1} \cdot i_{x_o} \cdot i_{x_1} \cdots i_{x_t}$ and for every $1 \leq i \leq t$

$$\alpha_{i+1}(\underline{x}) \triangleq E_{x_{(i+1)|_{(t+1)}}} \cdot i_{x_o} \cdot i_{x_1} \cdots i_{x_t} \cdot d_{x_t} \cdots d_{x_1} \cdot d_{x_o} \cdot D_{x_i},$$

where $n|_m$ denotes the reduction of $n$ modulo $m$. Note that the protocol, which is initiated by $x_o$, allows each of the $x_i$'s to read the initial message.

Using induction, one can show that in each step of the protocol a user decrypts the message using his instance of the PKCS but transmits it encrypted by an instance of one of the original users.

Thus, whenever a saboteur eavesdrops a function of the initial message, it is encoded by one of the original users. Thus, the protocol is secure.

Q.E.D.

As we shall see, the use of ordered cancellation is essential.

Lemma 8: For $\Sigma$ which consists of operators with unordered cancellation rules (i.e. if $\sigma\tau \equiv \lambda$ is a cancellation rule then so is $\tau\sigma \equiv \lambda$) there is no 3-party MRP.

The proof is given in Appendix F.

The lemma holds even if the words of the protocol must not be in their reduced forms. The proof of Lemma 8 can be extended to any number of participants thus proving the following:

Theorem 7: For $\Sigma$ which consists of operators with unordered cancellation rules there exists no MRP.

We define an echo-protocol to be a secure two-party ping-pong protocol in which the initiator can read the initial message after his counterpart has read it. Note that $P(x,y) = \{\alpha_i(x,y)\}_{i=1}^{2}$, where $\alpha_1(x,y) \triangleq E_y \cdot i_x$ and $\alpha_2(x,y) \triangleq E_x \cdot d_x \cdot D_y$ is an echo-protocol. However, using a modification of the claim, used (in Appendix F) to prove Lemma 8, one can prove the following:

Theorem 8: For $\Sigma$ which consists of operators with unordered cancellation rules there exists no echo-protocol.

## APPENDIX A: ON THE ASSUMPTION THAT EXACTLY p USERS LEGITIMATELY USE A p-PARTY PROTOCOL

We consider it very natural to assume that if a protocol is defined for p users then it should be used by p different users. (For example, consider a 3-party ping-pong protocol which is a simulation of certified-mail from sender to addressee via the post-office. Clearly, an honest user will participate in an instance of this protocol only if it is played by 3 different users.) Also note that the "correct" way to introduce protocols for a variable number of participants is to introduce a family of protocols such that each protocol, $P_i$, is defined for a fixed number of participants, $p_i$, and should be used by $p_i$ different users.

However, if this assumption is not made and it is assumed that an honest user is willing to participate in any instance of the protocol (even if it is played by less than p users), then the definition of insecurity differs from the definition given in Section 2 and proceeds as follows:

A p-party ping-pong protocol $P \stackrel{\Delta}{=} \{\alpha_j(\underline{x})\}_{j=1}^{\ell}$ is <u>insecure</u> if there exist a set S and a string $\gamma$ such that

$$\gamma \in ((\bigcup_{z\in S} \Sigma_z) \cup \{\alpha_j(\underline{b}): 1 \leq j \leq \ell,\ B \subseteq (A \cup S),\ |B| \leq p\})^*$$

and $\overline{\gamma\cdot\alpha_1(\underline{a})} = \lambda$. Note that the choice of $\underline{a}$ (as long as $|A| \leq p$) is immaterial.

Under this definition, if a protocol is insecure then a single saboteur suffices to demonstrate it. (Simply replace all saboteurs in $\gamma$ by the saboteur s.) Thus, for every fixed p the security of p-party ping-pong protocols can be tested in time $O(n^3)$ and space $O(n^2)$, using the technics described in DEK. However, Yair Itzhaik [I] showed that, if the number of participants is part of the security-problem's input then the problem is NP-Complete.

## APPENDIX B: PROOF OF LEMMA 6

Our goal (Lemma B6) is to show that if $s$ is a t-string and $G_s$ is a c-chromatic then $c \leqslant 3(t-1)+2$.

Definitions:

Let $s$ be a t-string, we say that a vertex, v, of the graph $G_s$ appears in a word, w, (of the string) if the variable associated with v occurs in w. Also, an edge, (u,v), appears in a word, w, if both its endpoints, u and v, appear in w. We say that a vertex [an edge] of $G_s$ appears in a subset of the words of s if it appears in a word of this subset.

We say that a line carries a vertex, v, between two words, $w_1$ and $w_2$, if there is a pair* of occurrences of the variable associated with v such that one element of the pair is in $w_1$ while the other is in $w_2$. We define a path of lines (pol) which carries a vertex, v, between two words, $w_1$ and $w_2$, recursively as follows:

(1) If there is a line which carries v between $w_1$ and $w_2$ then there is a pol which carries v between $w_1$ and $w_2$.

(2) If there is a pol which carries v between $w_1$ and $w_2$ and a pol which carries v between $w_2$ and $w_3$, then there is a

* See the definition of a pair in Section 3, following the proof of Theorem 1.

pol which carries $v$ between $w_1$ and $w_3$.
(Note that a pol which carries $v$ between $w_1$ and $w_2$ corresponds to a route between an occurrence in $w_1$, of the variable associated with $v$, and an occurrence in $w_2$ of this variable.)
Note that according to the definition of t-strings, there exists a route between every two occurrences of a variable. Thus, if a vertex, $v$, appears in two words of a t-string, then there is a pol which carries $v$ between them. We refer to this property as the vertex unity.

Occasionally it will be convenient to view a t-string, $s$, as a closed (circular) one rather than viewing $s$ as open. Modulo this convention the string $s = s's''$, where $s'$ and $s''$ are substrings of $s$, is congruent to $s''s'$; clearly the SA problems for $s's''$ and $s''s'$ are identical.

We denote by $L(s;s_1,w_1,s_2,w_2)$ a line between $w_1$ and $w_2$, where $s = s_1w_1s_2w_2$. We say that this line separates $s$ into two substrings, $s_1$ and $s_2$.

Lemma B1: Let $s$ be a t-string and $L(s;s_1,w_1,s_2,w_2)$ be a line in $s$. All the vertices which appear in both $s_1$ and $s_2$, appear in either $w_1$ or $w_2$.

Proof: Let $v$ be a vertex which appears in both $s_1$ and $s_2$. Let $w_1'$ $[w_2']$ be a word of $s_1$ $[s_2]$ such that $v$ appears in $w_1'$ $[w_2']$. By the vertex unity, there is a path of lines which carries $v$ between $w_1'$ and $w_2'$. Note that there is no line between a word of $s_1$ and a word of $s_2$. Therefore, there is an $i \in \{1,2\}$ such that there is a path of lines which carries $v$

between $w_1'$ and $w_i$ and a path of line between $w_i$ and $w_2'$. Thus, $v$ appears in $w_i$ and the lemma follows.

□

Definitions: We say that a t-string, $s$, is a c-requiring t-string if the chromatic number of $G_s$ is at least $c$ and $c \geq 3(t-1)+1$. We say that a t-string, $s$, is a minimum c-requiring t-string if it satisfies the following two conditions:

(1) $s$ is a c-requiring t-string

(2) $s$ has the minimum number of variables w.r.t. (1).

Lemma B2: Let $s$ be a minimum c-requiring t-string, and consider a line $L(s;s_1,w_1,s_2,w_2)$ in it. Let $W$ be the set of vertices which appear in $w_1$ or $w_2$. Let $S_1$ [$S_2$] be the set of vertices which appear in $s_1$ [$s_2$]. Either $S_1 \subseteq W$ or $S_2 \subseteq W$.

Proof: Assume, on the contrary to the statement of the lemma, that neither $S_1 \subseteq W$ nor $S_2 \subseteq W$. Lemma B1 implies that $S_1 \cup W \subsetneq V_s$ and $S_2 \cup W \subsetneq V_s$.

Let us denote by $v_o$ the vertex which is carried by the line L. For $1 \leq i \leq 2$, denote by $W_i$ the set of vertices which appear in $w_i$. Denote a typical element of $W_1 - \{v_o\}$, by the letter $u$, and of $W_2 - \{v_o\}$, by $v$; $W_1 - \{v_o\} = \{u_j: 1 \leq j \leq q_1\}$ and $W_2 - \{v_o\} = \{v_j: 1 \leq j \leq q_2\}$.
Clearly, each set may be empty and $q_i \leq t-1$ for $1 \leq i \leq 2$.

Denote by $s_1''$ [$s_2''$] the string which results from $w_2 s_1 w_1$ [$w_1 s_2 w_2$] by omitting the occurrences (of variables) which are unpaired in this string. (Note that these occurrences are paired in s with occurrences in $s_2$ [$s_1$]. Also note that $s_1''$ [$s_2''$] can be unlinked since occurrences of a variable in $w_1$ and $w_2$ may have been linked in $s$ (only) through $s_2$ [$s_1$].)

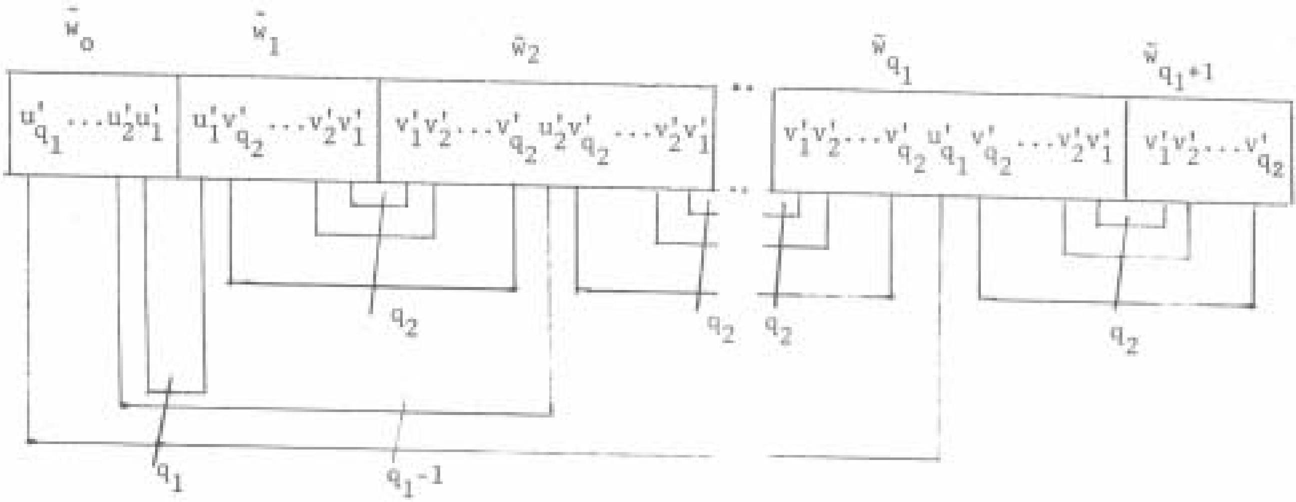
Consider the following two cases:

Case 1: $q_1 \cdot q_2 = 0$

Denote $s_i' \triangleq s_i''$, for $1 \le i \le 2$. (Note that $s_i''$ is linked.)

Case 2: $q_1 \cdot q_2 > 0$

Consider the t-string $s_o$ presented in Figure 3.




(u' [v'] denotes the variable associated with u [v])

Figure 3

Note that if $u_i = v_j$ then the variable associated with $v_j$ occurs in $\tilde{w}_o$ as well as in $\tilde{w}_{q_1+1}$ and there exists a route, which goes through $\tilde{w}_i, \tilde{w}_{i+1}, \ldots, \tilde{w}_{q_1}$ between these occurrences. On the other hand, if for every $1 \le j \le q_2$, $u_i \ne v_j$ then the vertices $u_i, v_1, v_2, \ldots, v_{q_2}$ appear in $\tilde{w}_i$ and therefore are assigned different colors in every coloring of $G_{s_o}$.

Denote by $s'_o$ the string which results from the reflection of $s_o$.

Denote by $s'_1$ $[s'_2]$ the string which results from $s''_1 w_1 s_o w_2$ $[s'_o w_1 s''_2 w_2]$ by merging $w_1$, $\tilde{w}_o$ into one word and merging $\tilde{w}_{q_2+1}$, $w_2$ into one word.

Note that $s'_1$ $[s'_2]$, constructed in each of the cases, is a t-string. Also note that $|V_{s'_1}| < |V_s|$ $[|V_{s'_2}| < |V_s|]$. This would lead, as we will show below, to a contradiction implying that the statement of the lemma does hold.

First, note that $s'_1$ $[s'_2]$ is not a c-requiring t-string. (Otherwise our assumption that $s$ is a minimum c-requiring t-string is contradicted.) Thus, for $1 \le i \le 2$, there exists a coloring of $G_{s'_i}$, $f_i$, which uses less than $c$ colors.

Note that in every coloring of $G_{s'_i}$, the vertices of $W$ are assigned different colors. (Since every coloring of $G_{s'_i}$ induces a coloring of $G_{s_o}$ and in every coloring of $G_{s_o}$ the vertices of $W$ are assigned different colors.) Also note that the intersection of the domains of $f_1$ and $f_2$ is $W$. Thus, $f_1$ and $f_2$ can be merged, consistently, to yield a coloring of $G_s$ with less than $c$ colors. However, this contradicts our assumption that $s$ is a c-requiring t-string.

□

Lemma B3: Let $L(s;s_1,w_1,s_2,w_2)$ be a line in a minimum c-requiring t-string, s. There exists a vertex (in $G_s$) which does not appear in $w_1$ or $w_2$.

Proof: Assume on the contrary that all the vertices of $G_s$ appear in $w_1$ and $w_2$. This yields $c \leq |V_s| \leq 2t-1$, in contradiction to $c \geq 3(t-1)+1$ (condition (1) of the definition of a minimum c-requiring t-string), since $t > 1$.

□

We are now ready to define the line structure of a minimum c-requiring t-string s. Consider a line between $w_1$ and $w_2$ denoted $L(s;s_1,w_1,s_2,w_2)$. By Lemmas B2 and B3 there exist a unique $i \in \{1,2\}$ such that every vertex which appears in $s_i$, appears either in $w_1$ or in $w_2$. Draw the line from $w_1$ to $w_2$ below this $s_i$; the words of $s_i$ are said to be above this line.

Let us show that it is possible to draw the lines according to the definition of the line structure so that no two lines cross. Note that a line drawn between the occurrences $\theta_1$ and $\theta_2$ crosses the line drawn between the occurrences $\theta_3$ and $\theta_4$ iff the first line is drawn below $\theta_3$ and $\theta_4$ and the latter is drawn below $\theta_1$ and $\theta_2$.

Lemma B4: Let s be a minimum c-requiring t-string. No lines cross in the line structure of s.

Proof: Assume that a line between the occurrences $\theta_1$ and $\theta_2$ is drawn below the occurrences $\theta_3$ and $\theta_4$ and that there is a line between $\theta_3$ and $\theta_4$. Let $w_1$ [$w_2$] be the word in which $\theta_1$ [$\theta_2$] occurs, and $L(s;s_1,w_1,s_2,w_2)$ be the line between $\theta_1$ and $\theta_2$.

Assume, with no loss of generality, that the line between $\theta_1$ and $\theta_2$ is drawn below $s_1$. By the construction of the line structure, there exists a vertex which appears in $s_2$ but not in $w_2s_1w_1$. Note that $\theta_3$ and $\theta_4$ occur in $w_2s_1w_1$ and therefore the words which contain them ($w_3$ and $w_4$, respectively) are contained in $w_2s_1w_1$. Thus, there exists a vertex which appears in $s_2$ but neither appears in $w_3$, nor in $w_4$. By the construction of the line structure, the line between $\theta_3$ and $\theta_4$ is drawn below a substring of $s_1$ and therefore is not drawn below $\theta_1$ or $\theta_2$.

□

We say that the line between $\theta_1$ and $\theta_2$ <u>covers</u> the line between $\theta_3$ and $\theta_4$ if the line between $\theta_1$ and $\theta_2$ is drawn below $\theta_3$ and $\theta_4$.

Let us now define the <u>star-line structure</u> of a minimum c-requiring t-string s. Consider the set of "out-most" lines (i.e. the lines that are not covered by any line) in the line structure of s. The lines in this set are called <u>star-lines</u> ($s\ell$). The words on the end of a star-line are called its <u>star-words</u> (sw) and the words above a star-line are called its <u>word-interval</u>. Note that a word-interval may be empty. (Fig. 4 shows the line structure and the star-line structure of the 2-string presented in Lemma 3.)

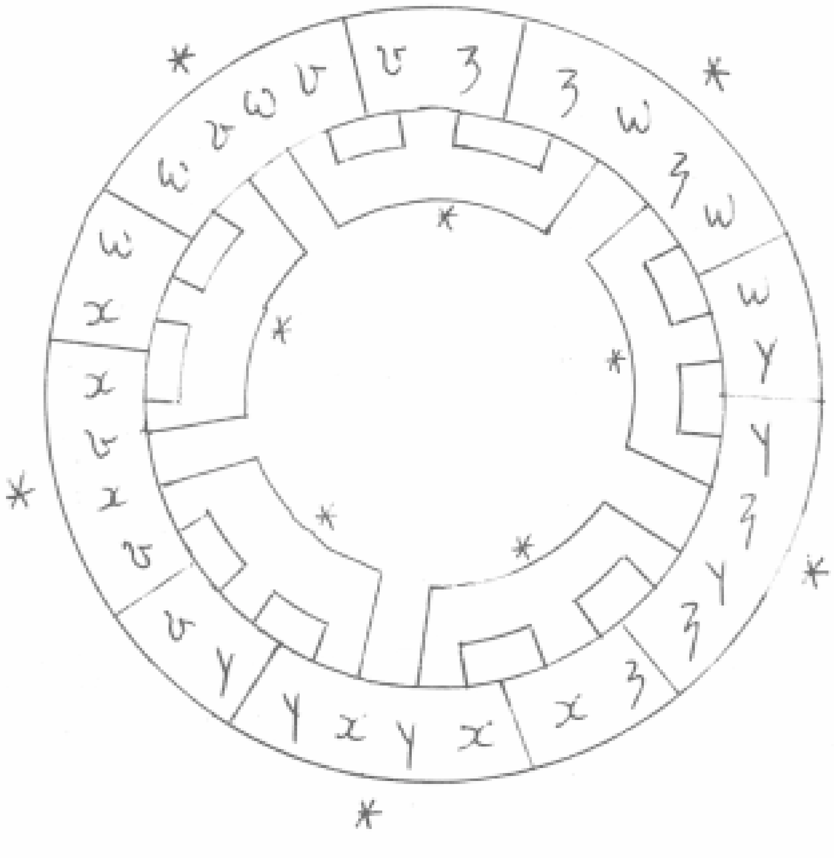


Figure 4

The following lemma is the core of the entire proof (of Lemma 6):

Lemma B5: Let s be a minimum c-requiring t-string, such that the length of s (i.e. the number of occurrences in s) is minimal. $|E_s| \leq (1.5\cdot t-1)\cdot|V_s| + 0.5\cdot t$.

Proof: The vertex unity property is instrumental in this proof; it states that if a vertex appears in two words of s then there is a path of lines between them.

The following facts, about the star-line structure of s, are of interest:

Fact 1: Each word is either a sw or belongs to some word-interval. (Otherwise, by the vertex unity property, the vertices which appear in such a word do not appear in other words of s. Thus, the word can be omitted from s resulting in a t-string s' such that s' is c-requiring and $|V_{s'}| < |V_s|$, contradicting our assumption that s is a minimum c-requiring t-string.)

Fact 2: There is at most one pair of adjacent sw's with no sℓ between them. (Otherwise, s can be divided into two variable-disjoint t-strings, such that at least one of them is a c-requiring t-string, again contradicting our assumption that s is a minimum c-requiring t-string.)

Fact 3: The vertices which appear in the word interval of a sℓ appear at least in one of its sw's. (By the construction of the line structure.)

Fact 4: The set of vertices which appear in the sw's is $V_s$. (By Facts 1 and 3.)

We say that a vertex v is transferred through the star-line L if the following hold: (1) There is a path of lines which carries v between the star-words of L. (2) This path of lines passes through words of L's word-interval only.

We say that a star-line is of multiplicity k if k vertices are transferred through it. If there is a pair of adjacent star-words with no star-line between them, then we constract a dummy star-line between them and say that it is of multiplicity zero.

Fact 5: There is no vertex which is transferred through all the star-lines. (Assume on the contrary that a vertex, v, is tranferred through all star-lines. Thus, there exists a closed route, among occurrences of the variable associated with v, going through all star-words. There are two adjacent occurrences on this route which are not in the same word, i.e. constitute a pair. The string, which results from s by omitting this pair, is also a minimum c-requiring t-string; however, it is shorter than s and thus contradicts our assumption that s is of minimal length.)

Denote by n the number of star-words. Beginning at an arbitrary sw and going clockwise, we number the sw's from 0 to n-1. Denote by $w_i$ the i-th sw and by $t_i$ the number of vertices which appear in it. Denote by $k_i$ the multiplicity of the s$\ell$ between $w_i$ and $w_{i+1}$, where indices are considered modulo n.

Let A(v) denote the number of sw's in which the vertex v appears. Using the vertex unity property (and Fact 4), we get

Fact 6: $\sum_{v \in V_s} (A(v)-1) \leq \sum_{i=0}^{n-1} k_i$.

Noting that $\sum_{v \in V_s} A(v) = \sum_{i=0}^{n-1} t_i$, we get

Fact 7: $\sum_{i=0}^{n-1} (t_i - k_i) \leq |V_s|$.

Let us denote by $e_I$ the number of edges which appear in word-intervals but not in star-words. (We remind the reader that an edge appears in a set of words if both its end-points appear together in some word of the set.)

Fact 8: $e_I \leq \sum_{i=0}^{n-1} (t_i - k_i)(t_{i+1} - k_i)$.

(Note that the vertices which are transferred by the $s\ell$ between $w_i$ and $w_{i+1}$ appear in both sw's. Also note that, by Fact 3 and the definition of $e_I$, a necessary condition for an edge to be counted in $e_I$ is that one of its endpoints appears in $w_i$ but not in $w_{i+1}$ and the other appears in $w_{i+1}$ but not in $w_i$.)

Denote by $e_{sw}$ the number of edges which appear in star-words. By Fact 5, for every vertex there exists a star-line through which it is not transferred. Also note, that by the vertex unity concept there is a single interval of star-lines through which a vertex is transferred. We say that the vertex $v$ is <u>born</u> in $w_{i+1}$ if $v$ appears in $w_{i+1}$ and is not transferred through the $s\ell$ between $w_i$ and $w_{i+1}$. The edge $(u,v)$ is counted at $w_{i+1}$ if $v$ is born in $w_{i+1}$ and $u$ (also) appears in it.

Fact 9: The number of edges counted at $w_{i+1}$ is bounded from above by $(t_{i+1}-k_i)\cdot k_i + \binom{t_{i+1}-k_i}{2}$. (i.e. the sum of the number of edges between vertices which are born in $w_{i+1}$ and vertices which appear in $w_{i+1}$ but are not born there and the number of edges between vertices which are born in $w_{i+1}$.)

Elaborating Fact 9, we get

Fact 10: $e_{sw} \leq \frac{1}{2} \sum_{i=0}^{n-1} ((t_{i+1}^2 - k_i^2) - (t_{i+1} - k_i)) = \frac{1}{2} \sum_{i=0}^{n-1} (t_i^2 - k_i^2 - (t_i - k_i))$.

Combining Facts 8, 10 and noting that $t_i \leq t$, we get:

Fact 11: $|E_s| \leq e_I + e_{sw} \leq$

$$\sum_{i=0}^{n-1} (t_i - k_i)((t_{i+1} - k_i) + \frac{1}{2}(t_i + k_i - 1)) \leq \sum_{i=0}^{n-1} (t_i - k_i)\left(\frac{3}{2}\cdot t - \frac{1}{2}\cdot k_i - \frac{1}{2}\right).$$

By Fact 2 there is at most one $i$ for which $k_i = 0$. Thus, we get

Fact 12: $|E_s| \leq \sum_{i=0}^{n-1} (t_i - k_i)\cdot\left(\frac{3}{2}\cdot t - \frac{1}{2}\cdot 1 - \frac{1}{2}\right) + \frac{1}{2}\cdot t$.

Using Fact 7 we get

$|E_s| \leq (1.5\cdot t - 1)\cdot|V_s| + 0.5\cdot t$, and the lemma follows.

□

<u>Lemma B6</u>: If $s$ is a t-string and $G_s$ is c-chromatic then $c \leq 3(t-1)+2$.

<u>Proof</u>: Assume $c > 3(t-1)+2$.

Consider a minimum c-requiring t-string with minimal length, hereafter denoted $s'$. Denote by k the chormatic number of $G_{s'}$. Note that $k \geq c$. By Lemma B5, $|E_{s'}| \leq (1.5\cdot t-1)\cdot|V_{s'}| + 0.5\cdot t$. Denote by $d_{min}$ [$d_{av}$] the minimum [average] degree of a vertex in $G_{s'}$. Obviously,

$$d_{min} \leq d_{av} = \frac{2\cdot|E_{s'}|}{|V_{s'}|} \leq 3t-2 + \frac{t}{|V_{s'}|} .$$

Since $c > 3(t-1)+2$, $|V_{s'}| \geq k \geq c > t$ holds. Thus $(t/|V_{s'}|)$ is a fraction implying that $d_{min} \leq 3t-2$. Note that $G_{s'}$ is critical* and therefore $d_{min} \geq k-1 \geq c-1$. Thus, $c-1 \leq 3t-2$ and the lemma follows. □

---

* A k-chromatic graph is called critical if the deletion of any vertex reduces the chromatic number of the graph.
Note that the minimum degree of a vertex in a critical k-chromatic graph is at least k-1.

## APPENDIX C: PROOF OF LEMMA 7

The 3XC problem is defined as follows: Given a set, $U = \{e_i\}_{i=1}^{3n}$, and a collection, $S = \{s_j\}_{j=1}^{m}$, of three-element subsets of U, determine whether there exists a subcollection of S such that every element of U appears exactly in one subset of the subcollection.

The 3XC problem is known to be NP-Complete [GJ]. A restricted version of this problem (hereafter denoted by r3XC), in which, in the input, every element is restricted to appear in at most three subsets, is also known to be NP-Complete [GJ]. One can easily prove this by reducing 3XC to r3XC as follows: Replace every element, which occurs in more than three subsets, of the 3XC instance by the following construction (see Fig. 5) resulting in a r3XC instance.

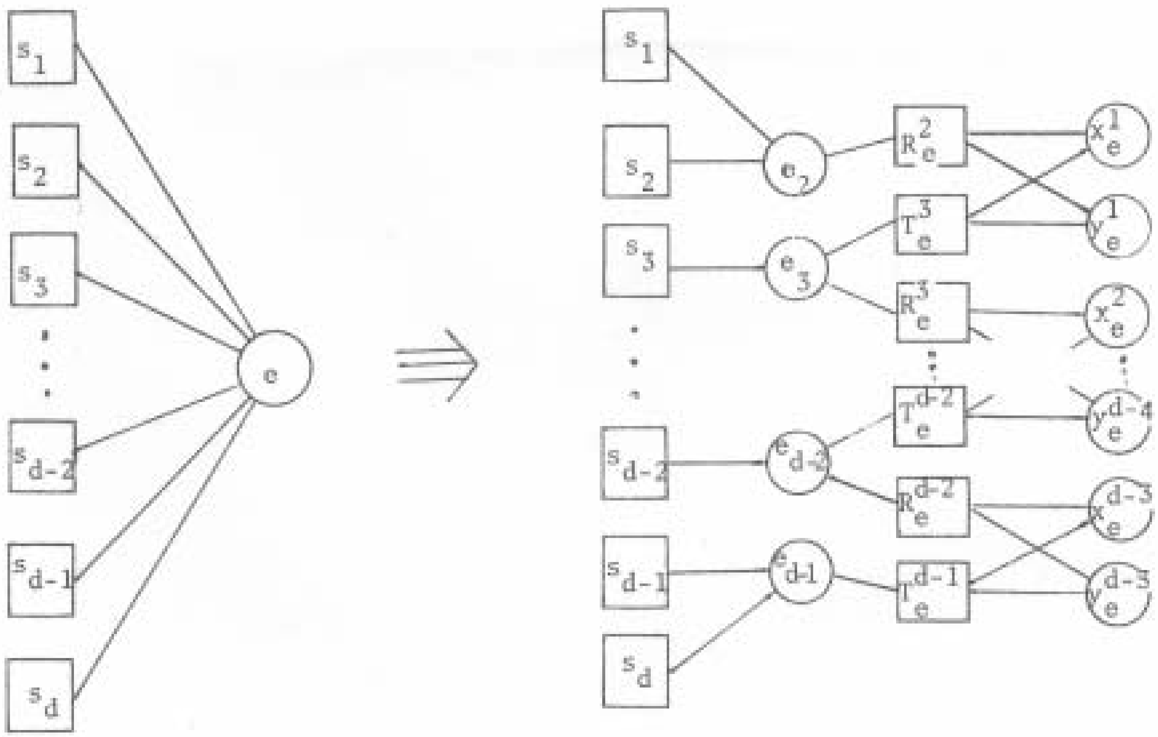



Figure 5

(Fig. 5 demonstrates the construction for an element, denoted e, which occurs in $d > 3$ subsets, denoted $s_1, s_2, \ldots s_d$.) Note that there is a solution of 3XC iff there is one of the corresponding r3XC.

In order to prove the lemma let us reduce r3XC to R3XC. First let us apply the following simplification process to the instance of the r3XC problem: If there is an element which appears only in one subset, omit the subset and its elements from the instance and apply the process to the result. Note that in each "phase" of this process the subset which is omitted must participate in every exact cover of the original instance of r3XC. When the process terminates we either get an empty instance or get a r3XC instance in which each element appears in two or three subsets. (Note that there is an exact cover in the simplified r3XC iff there is an exact cover in the original r3XC.) Partition the elements, which appear in two subsets each, into triples. For each triple apply a construction, as shown in Fig. 6 for the triple $e_1, e_2, e_3$. This transformation yields a R3XC instance.

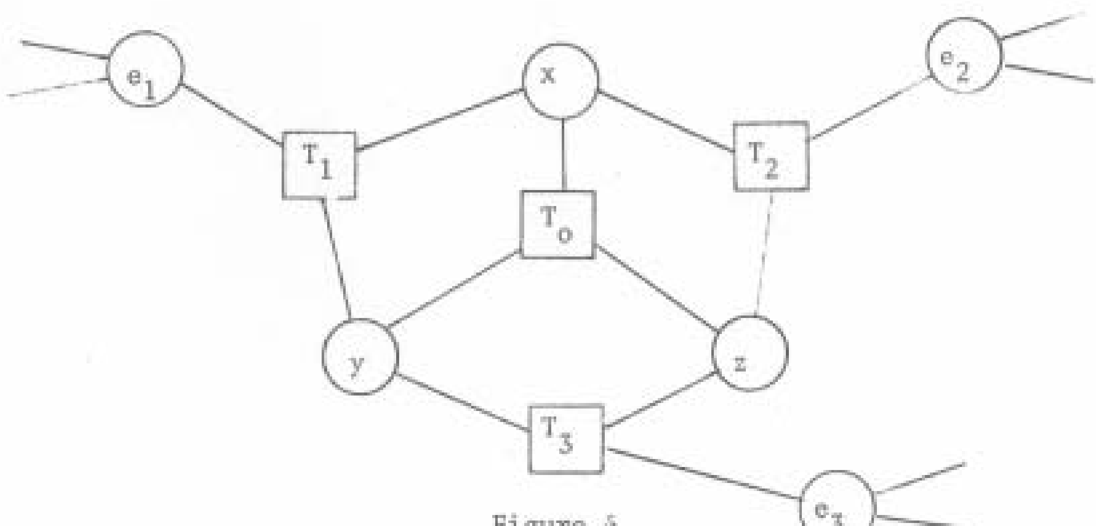



Figure 6

Note that the subset $T_o$ must participate in every exact cover of the R3XC instance and that the subsets $T_1$, $T_2$ and $T_3$ are never used. Thus, there exists an exact cover of the r3XC instance iff there exists an exact cover of the corresponding R3XC instance.

## APPENDIX D: REASONS TO BELIEVE THAT SMPP ∉ NP

Theorem D1: There exists a family of two-party ping-pong protocols such that each is insecure and its shortest insecure string is of length exponential in the length of the protocol.

Proof: Consider the following family: for every $n \geq 1$ $P_n \triangleq \{\alpha_i(x,y)\}_{i=1}^{n+1}$, where $\alpha_1 \triangleq E_x^{(1)}$, $\alpha_{n+1} \triangleq D_x^{(n)}$ and for every $1 < i \leq n$ $\alpha_i \triangleq E_x^{(i)} E_x^{(i)} D_x^{(i-1)}$. The following two claims can be proven by induction on i: (1) For every $a \in N$, a saboteur can effect $D_a^{(i)}$ on any message. (Here the induction is from n to 1.) (2) Let $\underline{a} = (a,b)$, $D_a^{(i)}$ appears at least $2^{i-1}$ times in any insecurity string of $P_n(\underline{a})$. (Here the induction is from 1 to n.)

Q.E.D.

Note that the theorem holds even if $\Sigma$ is restricted to $\{i_x, d_x, E_x, D_x : x \in N\}$. (Use $E_x^{(i)} = E_x(i_x)^i$ and $D_x^{(i)} = (d_x)^i D_x$ to modify the definition of $P_n$, and prove that $i_a(d_a)^j D_a$ appears at least $2^{j-1}$ times in any insecurity string.)

Corollary D1: SMPP cannot be solved in polynomial time by guessing an insecurity string and checking it.

Tha above is no reason to believe that SMPP ∉ NP, because we know that there is a much more effective way to check insecurity, namely the algorithm described in DEK. One may think that for every insecure protocol one can guess a polynomial number of instances of each protocol word and run the collapsing algorithm described in DEK on the partial automaton. This may work only if for every insecure protocol there is an insecurity string in which every protocol word occurs only in a polynomial number of different instances. We shall show that this condition does not hold, namely:

Theorem D2: There exists a family of multi-party ping-pong protocols such that each is insecure and each of its insecurity strings contains an exponential (in the number of participants) different instances of a certain protocol word.

Before proving the theorem let us consider the following family: for every $n \geq 3$, $P_n \triangleq \{\alpha_i^{(n)}(x_o,x_1,x_2,\ldots,x_n): 1 \leq i \leq 3\}$ is a $(n+1)$-party ping-pong protocol, where

$$\alpha_1^{(n)}(\underline{x}) \triangleq E_{x_o}\cdot(i_{x_2}\cdot i_{x_3}\cdots i_{x_n}\cdot i_{x_1})\cdot(i_{x_1}\cdot i_{x_2}\cdot i_{x_3}\cdots i_{x_n}) ,$$

$$\alpha_2^{(n)}(\underline{x}) \triangleq E_{x_o}\cdot(i_{x_2}\cdot i_{x_3}\cdots i_{x_n}\cdot i_{x_1})\cdot(d_{x_n}\cdots d_{x_3}\cdot d_{x_2}\cdot d_{x_1})\cdot D_{x_o}$$

and

$$\alpha_3^{(n)}(\underline{x}) \triangleq (d_{x_n}\cdots d_{x_2}\cdot d_{x_1})\cdot(d_{x_n}\cdots d_{x_2}\cdot d_{x_1})\cdot D_{x_o} .$$

Let $\theta_n$ be the following transformation $\theta_n((z_o,z_1,z_2,z_3,\ldots,z_n)) = (z_o,z_2,z_3,\ldots,z_n,z_1)$, for every $(n+1)$ vector $\underline{z}$. $\theta_n^j(\underline{z})$ is defined as $\theta_n\cdot(\theta_n^{j-1}(\underline{z}))$ if $j > 1$ and $\theta_n(\underline{z})$ if $j = 1$. Note that $\alpha_3(\underline{a})\cdot\alpha_2(\theta_n^{n-1}(\underline{a}))\cdots\alpha_2(\theta_n^2(\underline{a}))\cdot\alpha_2(\theta_n^1(\underline{a}))\cdot\alpha_1(\underline{a})$ is an insecurity string of $P_n$, where $\underline{a} \triangleq (a_o,a_1,\ldots,a_n)$. Note that every insecurity string of $P_n$ must contain an instance of $\alpha_3^{(n)}$ and at least $(n-1)$ different instances of $\alpha_2^{(n)}$. [Note first that $\alpha_3^{(n)}$ is the only protocol word which contains $D_{x_o}$ and does not contain $E_{x_o}$. However, an instance of $\alpha_3^{(n)}$ can be applied to $E_{a_o}$ only if $(i_{b_1} \cdot i_{b_2} \cdots i_{b_n}) \cdot (i_{b_1} \cdot i_{b_2} \cdots i_{b_n})$ appears on $E_{a_o}$'s r.h.s. . Note that in $\alpha_1^{(n)}(a)$, $(i_{a_2} \cdot i_{a_3} \cdots i_{a_n} \cdot i_{a_1}) \cdot (i_{a_1} \cdot i_{a_2} \cdots i_{a_n})$ appears on $E_{a_o}$'s r.h.s. and thus the only instance of a protocol word which can be applied to it is $\alpha_2^{(n)}(\theta_n^1(\underline{a}))$, resulting in $E_{a_o}(i_{a_3} \cdot i_{a_4} \cdots i_{a_1} \cdot i_{a_2}) \cdot (i_{a_1} \cdot i_{a_2} \cdots i_{a_n})$. Similar arguments force the use of $\alpha_2^{(n)}(\theta_n^2(\underline{a})), \ldots, \alpha_2^{(n)}(\theta_n^{n-1}(\underline{a}))$ in the insecurity string.]

Generalizing this construction we can prove Theorem D2.

Proof (of Theorem D2): For every $n \geq 1$, define $\{p_i^{(n)}\}_{i=1}^{n}$ to be the set of the first $n$ primes which are greater than $n$. Denote $q_j^{(n)} \triangleq \sum_{i=1}^{j-1} p_i^{(n)}$, for $1 < j \leq n+1$ and $q_1^{(n)} \triangleq 0$. Also define for $1 \leq j \leq n$

$$I_j^{(n)} \triangleq i_{x_{q_j^{(n)}+1}} \cdot i_{x_{q_j^{(n)}+2}} \cdots i_{x_{q_j^{(n)}+p_j^{(n)}}} ,$$

$$\bar{I}_j^{(n)} \triangleq i_{x_{q_j^{(n)}+2}} \cdot i_{x_{q_j^{(n)}+3}} \cdots i_{x_{q_j^{(n)}+p_j^{(n)}}} \cdot i_{x_{q_j^{(n)}+1}} ,$$

and

$$C_j^{(n)} \triangleq d_{x_{q_j^{(n)}+p_j^{(n)}}} \cdots d_{x_{q_j^{(n)}+2}} \cdot d_{x_{q_j^{(n)}+1}} .$$

Consider the following family of protocols:

For every $n \geq 1$, $\mathbb{P}_n \triangleq \{\bar{\alpha}_i^{(n)}(x_o, x_1, \ldots x_{q_{n+1}^{(n)}}) : 1 \leq i \leq 3\}$ is a $(q_{n+1}^{(n)} + 1)$-party ping-pong protocol, where

$$\tilde{\alpha}_1^{(n)}(\underline{x}) \triangleq E_{x_o} \cdot \overset{+}{I}{}_1^{(n)} \cdot \overset{+}{I}{}_2^{(n)} \cdots \overset{+}{I}{}_n^{(n)} \cdot I_1^{(n)} \cdot I_2^{(n)} \cdots I_n^{(n)} \quad ,$$

$$\tilde{\alpha}_2^{(n)}(\underline{x}) \triangleq E_{x_o} \cdot \overset{+}{I}{}_1^{(n)} \cdot \overset{+}{I}{}_2^{(n)} \cdots \overset{+}{I}{}_n^{(n)} \cdot C_n^{(n)} \cdots C_2^{(n)} \cdot C_1^{(n)} \cdot D_{x_o}$$

and

$$\tilde{\alpha}_3^{(n)}(\underline{x}) \triangleq C_n^{(n)} \cdots C_2^{(n)} \cdot C_1^{(n)} \cdot C_n^{(n)} \cdots C_2^{(n)} \cdot C_1^{(n)} \cdot D_{x_o} \quad .$$

We claim that $\tilde{P}_n$ is insecure and every insecurity string of it must contain at least $(\prod_{i=1}^{n} p_i^{(n)})-1$ different instances of $\tilde{\alpha}_2^{(n)}$. (The proof of this claim is a generalization of the considerations applied to $P_n$.) Note that $\prod_{i=1}^{n} p_i(n)$ is exponential in $\sum_{i=1}^{n} p_i^{(n)}$ .

## APPENDIX E: AN ALGORITHM FOR FINDING THE LENGTH OF THE SHORTEST INSECURITY STRING

After constructing the automaton, use the following algorithm to construct the collapsing relation: Let $M$ be a $(s+1) \times (s+1)$ matrix the entries of which are the length of the shortest collapsing path between the nodes of the automaton. Let $Q$ be a priority queue the elements of which are triples of integers, such that if $(i,j,\delta)$ is an element of $Q$ then $i$ and $j$ are states of the automaton and $\delta$ is the length of a collapsing path from $i$ to $j$. If $\delta_1 < \delta_2$ then $(i_1,j_1,\delta_1)$ is prior to $(i_2,j_2,\delta_2)$ in $Q$. Let $M_p$ be a $(s+1) \times (s+1)$ matrix the entries of which are pointers to elements of $Q$. Let $M_s$ be a $(s+1) \times (s+1)$ matrix the entries of which are elements of the set $\{N,T,F\}$. During the execution of the algorithm, $M_s(i,j) = N$ denotes the case where no collapsing path has been found from $i$ to $j$; $M_s(i,j) = T$ denotes the case where such a path was found but it is still unknown whether it is the shortest collapsing path from $i$ to $j$; $M_x(i,j) = F$ denotes the case where a shortest collapsing path from $i$ to $j$ has been found. Let $M'_t$ $[M''_t]$ be a $(s+1) \times (s+1)$ matrix the entries of which are pairs of states [operators]. $M'_t$ and $M''_t$ will be used to reconstruct the shortest insecurity string after the algorithm stops.)

The algorithm proceeds as follows:

(0) For $0 \leq i \neq j \leq s$ do $M_s(i,j) := N$;

For $0 \leq i \leq s$ do Insert$((i,i,0),Q,(i,i),(\lambda,\lambda))$;

(1) While $Q \neq \emptyset$ do begin

(1.1) Delete the first triple, $(i,j,\delta)$, from $Q$;

$M(i,j) := \delta$; $M_s(i,j) := F$;

(1.2) For $0 \leq k \leq s$ do

If $M_s(j,k) = F$ then Update $(i,k,\delta+M(j,k),Q,(j,j),(\lambda,\lambda))$;

(1.3) For $0 \leq k \leq s$ do

If $M_s(k,i) = F$ then Update $(k,j,M(k,i)+\delta,Q,(i,i),(\lambda,\lambda))$;

(1.4) For all edges entering $i$ and all edges leaving $j$ do

If $k \overset{\sigma}{\to} i$ +, $j \overset{\tau}{\to} \ell$ + and $\sigma\tau \equiv \lambda$ ++ then

Update $(k,\ell,1+\delta+1,Q,(i,j),(\sigma,\tau))$

end;

where the procedures Insert and Update are defined as follows:

---

\+ $i \overset{\sigma}{\to} j$ denotes the edge going from $i$ to $j$ and labelled $\sigma$.

++ $\sigma\tau \equiv \lambda$ denotes the cancellation rule which reduces $\sigma\tau$ to $\lambda$.

```
Procedure  Insert ((i,j,δ),Q,(q,r),(σ,τ));
begin
    Insert the triple (i,j,δ) to Q and set M_p(i,j) to be a pointer
to the position of (i,j,δ) in Q;
M_s(i,j):=T;  M'_t(i,j):=(q,r); M''_t(i,j):=(σ,τ);
end
Procedure Update (i,j,δ,Q,(q,r),(σ,τ));
begin
    Case of M_s(i,j)
    N: Insert ((i,j,δ),Q,(q,r),(σ,τ));
    T: [M_p(i,j) points to (i,j,δ')]
       If δ' > δ  then begin
            Delete (i,j,δ') from Q;
            Insert ((i,j,δ),Q,(q,r),(σ,τ));
       end;
    F:;
    end
end
```

Note that a priority queue can be implemented such that inserting [deleting] an element costs $O(\log_2 q)$ operations, where $q$ is the maximum length of the queue. The space such an implementation requires is $O(q \log_2 q)$. Also note that the first element of the queue can be found in constant time. Note that $Q$ contains at most one triple such that the first component of the triple is $i$ and the second is $j$. Thus, the running time of the procedures Update and Insert is $O(\log s)$.

Note that by the construction of the automaton, for $1 \le i \le s$ there is a single edge entering $i$ and at most one edge leaving it.

Also, there are at most $(s+m)$ edges entering [leaving] state zero, where $m$ is the number of self-loops from 0 to 0. Thus, the loop in step (1.4) is executed at most $1\cdot(s+m)^2 + 2s\cdot(s+m) + s^2\cdot 1 < 4(s+m)^2$ times. Clearly, the loop in step (1.2) [(1.3)] is executed at most $s^2\cdot s$ times. Thus, the algorithm runs in time $O(n^3\log_2 n)$, where $n$ is the length of the protocol $(n > s+m)$. Note that the algorithm requires space $O(s^2\log_2 s)$.

To allow the construction of the shortest insecurity string run the following recursive procedure, using the matrices $M_t'$ and $M_t''$ built by the above algorithm:

```
Procedure Track(i,j);
begin
     (q,r):=M't(i,j);  (σ,τ):=M"t(i,j);
     If σ = λ then begin  [Note that q = r, q ≠ i and q ≠ j.]
                   Track(i,q); Write  <q>; Track(q,j);
               end
               else if  q = r then Write   σ < q > τ
                                   else begin
                                          Write σ<q>; Track(q,r); Write <r>τ;
                                   end
end
```

Note that Track(0,1) outputs a shortest collapsing path specified by its nodes and the labels of its edges.

APPENDIX F: PROOF OF LEMMA 8

Assume, on the contrary, that such a protocol, $P(x,y,z) \triangleq \{\alpha_i(x,y,z)\}_{i=1}^{\ell}$, exists. Assume with no loss of generality that $x$ is the initiator of $P(x,y,z)$ and $y$ is the first user who reads the initial message, and that he first reads it while applying $\alpha_r$. Assume that $z$ first reads the initial message while applying $\alpha_t$ (obviously $r < t$).

Note that since the cancellation rules are unordered, the inverse of an operator can be defined. We denote the inverse of $\sigma$ by $\sigma^{-1}$. Note that $\overline{\sigma \cdot \sigma^{-1}} = \overline{\sigma^{-1} \cdot \sigma} = \lambda$.

Let $\alpha_r(x,y,z) = \beta_2(x,y,z) \cdot \beta_1(x,y,z)$ such that $y$ reads the initial message after applying $\beta_1(x,y,z)$. Note that
$\overline{\beta_1(x,y,z) \cdot \alpha_{r-1}(x,y,z) \cdots \alpha_2(x,y,z) \cdot \alpha_1(x,y,z)} = \lambda$.
Consider a cancellation pattern, $C$, which reduces the string $\beta_1(x,y,z) \cdot \alpha_{r-1}(x,y,z) \cdots \alpha_2(x,y,z) \cdot \alpha_1(x,y,z)$ to $\lambda$.
(Note that $C$ is a sequence of pairs of operators such that:

(1) The elements of the first pair cancel each other and are adjacent in the original string.

(2) The elements of the j-th pair cancel each other in the string which results from the original string after applying the first $(j-1)$ cancellations of C.

(3) The string which results from the original string, after applying all the pairs of C, is $\lambda$.)

We define, recursively, a <u>route</u> between $\alpha_i(x,y,z)$ and $\alpha_j(x,y,z)$, with respect to the cancellation pattern C, as follows:

(1) Both $\alpha_i(x,y,z)$ and $\alpha_j(x,y,z)$ are applied by $x$ and there is an operator in $\alpha_j$ and an operator in $\alpha_i$ such that these

operators cancel each other in C (i.e. constitute a pair of C).

(2) There is a protocol word, $\alpha_k(x,y,z)$, such that there exist a route between $\alpha_i$ and $\alpha_k$ and a route between $\alpha_k$ and $\alpha_j$.

We call k the <u>boundary</u> of P(x,y,z), with respect to C, if k is the greatest integer such that there exists a route between $\alpha_1(x,y,z)$ and $\alpha_k(x,y,z)$. Note that $k < r$, since $\alpha_r(x,y,z)$ is applied by y and if $i < r < j$ then there exists no pair in C such that its first element is in $\alpha_i(x,y,z)$ and its second in $\alpha_j(x,y,z)$. Let k be the boundary of P(x,y,z), apply C to $\alpha_k(x,y,z) \cdots \alpha_2(x,y,z) \cdot \alpha_1(x,y,z)$ and denote the result by $\gamma_1(x,y,z)$. (By applying C to $\alpha_k(x,y,z) \cdots \alpha_1(x,y,z)$ we mean applying only the cancellations between pairs of operators which are in $\alpha_k(x,y,z) \cdots \alpha_1(x,y,z)$. Note that this may differ from the reduced form of $\alpha_k(x,y,z) \cdots \alpha_1(x,y,z)$.) Note that $\gamma_1(x,y,z)$ contains only operators which occur in words that are applied by x. Thus, $\gamma_1(x,y,z)$ does not contain decryptions by either y or z. Also note that $\gamma_1(x,y,z)$ does not contain encryptions by x, since decryptions by x can only occur in words which are applied by x.

The following claim implies that P(x,y,z) is insecure and thus contradicts our assumption that P(x,y,z) is a 3-party MRP.

<u>Claim</u>: Two saboteurs, $s_1$ and $s_2$, can affect the inverse of any operator which occurs in $\gamma_1(a,b,c)$ $(= \sigma_n \cdots \sigma_2 \cdot \sigma_1)$ to any message.

<u>Proof</u>: Let $\sigma_q$ be an operator of $\gamma_1(a,b,c)$. If $\sigma_q$ is not an encryption then $\sigma_q^{-1} \in \Sigma_{s_1}$ and $s_1$ can apply it directly to any message. Otherwise, assume that $\sigma_q = E_v$, where $v \in \{b,c\}$ (since $E_a$ cannot occur in $\gamma_1(a,b,c)$). Define $P[b] \triangleq P(s_1,b,s_2)$ and

$P[c] \triangleq P(s_1,s_2,c)$. Note that $v$ plays in $P[v]$ the same role he plays in $P(a,b,c)$. Define $\alpha_i[b] \triangleq \alpha_i(s_1,b,s_2)$ $\gamma_1[b] \triangleq \gamma_1(s_1,b,s_2)$, $\alpha_i[c] \triangleq \alpha_i(s_1,s_2,c)$ and $\gamma_1[c] \triangleq \gamma_1(s_1,s_2,c)$. Let $\gamma_1[v] = \tilde{\sigma}_n \cdots \tilde{\sigma}_2 \cdot \tilde{\sigma}_1$. Note that $\tilde{\sigma}_q = \sigma_q$, since both are the same operator indexed by $v$. Also note that for $1 \le i \le n$ $\tilde{\sigma}_i \in \Sigma_{s_1}$, since $\gamma_1(x,y,z)$ contains no decryptions by $y$ or $z$. Note that for every message $M'$,

$$\beta_1[v] \cdot \alpha_{r-1}[v] \cdots \alpha_{k+1}[v] \cdot \tilde{\sigma}_n \cdots \tilde{\sigma}_{q+1}(M') = \tilde{\sigma}_1^{-1} \cdots \tilde{\sigma}_q^{-1}(M')$$

(i.e. $\beta_1[v] \cdot \alpha_{r-1}[v] \cdots \alpha_{k+1}[v] \cdot \tilde{\sigma}_n \cdots \tilde{\sigma}_{q+1} \equiv \tilde{\sigma}_1^{-1} \cdots \tilde{\sigma}_q^{-1}$).

(Since for any two strings of operators, $\delta_1$ and $\delta_2$, $\overline{\delta_1 \delta_2} = \lambda$ implies $\delta_1 \cdot \delta_2 \equiv \lambda$ and $\delta_1 \equiv \delta_2^{-1}$.)

Thus, $s_1$ and $s_2$ can get $\tilde{\sigma}_q^{-1}(M') = D_v(M')$ as follows:
$s_1$ initiates $P[v]$ and replaces the k-th transmission by $\tilde{\sigma}_n \cdots \tilde{\sigma}_{q+1}(M')$.
$s_2$ reads $\tilde{\sigma}_1^{-1} \cdots \tilde{\sigma}_{q-1}^{-1} \cdot \tilde{\sigma}_q^{-1}(\ )$ during the r-th step if $v = c$ and during the t-th step if $v = b$.
By applying $\tilde{\sigma}_{q-1} \cdots \tilde{\sigma}_1$ to what $s_2$ has read, the saboteurs get $\tilde{\sigma}_q^{-1}(M')$.

□

Having proven the claim, it is easy to see that $P(x,y,z)$ is insecure: Note that two saboteurs $s_1$ and $s_2$ can read the initial message, M, if they eavesdrop the k-th transmission of $P(a,b,c)$ (i.e. read $\gamma_1(a,b,c)(M)$) and play several instances of $P(x,y,z)$ with either b or c. Thus, Lemma 8 follows.

ACKNOWLEDGEMENT

The authors would like to thank Prof. Zvi Galil and Dr. Yair Itzhaik for discussions and constructive comments. The authors would like to thank Prof. Manuel Blum for his encouragement, his interest in the work and his support.

REFERENCES

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

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

[NS] R.M. Needham and M.D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers", Comm. ACM, Vol. 21, No.12, Dec. 1978, pp. 993-999.

[DY] D. Dolev and A.C. Yao, "On the Security of Public Key Protocols", to appear, IEEE Trans. on Inform. Th.

[R] B.K. Rosen, "Tree-Manipulating Systems and Church Rosser Theorems", JACM, Vol. 20, No.1, Jan. 1973, pp. 160-187.

[DEK] D. Dolev, S. Even and R.M. Karp, "On the Security of Ping-Pong Protocols", to appear in the Proceedings of CRYPTO 82.

[GJ] M.R. Garey and D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, W.H. Freeman and Co., 1979, pp. 221.

[P] E.L. Post, "A Variant of a Recursively Unsolvable Problem", Bulletin of the American Mathematical Society, 52, 1946, pp. 264-268.

[I] Y. Itzhaik, "A Protocol-Word Problem which is NP-Complete", private communication.