TECHNION - Israel Institute of Technology
Computer Science Department

ON THE SECURITY OF MULTI-PARTY PING-PONG PROTOCOLS

by

S. Even and O. Goldreich

Technical Report #285

June 1983

Part of this research was done while the authors visited the EECS Dept., UC-Berkeley. Supported by NSF grant No. MCS 82-04506. The first author was also supported by the Fund for the Promotion of Research at the Technion.

ABSTRACT

We define a p-party ping-pong protocol and its security problem, along the lines of Dolev and Yao's definitions for two-party ping-pong protocols.

In the case of two parties, it was assumed, with no loss of generality, that there exists a single saboteur in the net and the protocol was defined to be secure iff it was secure against the active interventions of one saboteur. We show that for more than 2 parties this assumption can no longer be made and that for p parties $3(p-2)+1$ is a lower bound on the number of saboteurs which should be considered for the security problem. On the other hand we establish a $3(p-2)+2$ upper bound on the number of saboteurs which should be considered. We conclude that for a fixed p, p-party ping-pong protocols can be tested for security in $O(n^3)$ time and $O(n^2)$ space, where n is the length of the protocol. We show that if p, the number of participants in the protocol, is part of the input then the security problem becomes NP-Hard. Relaxing the definition of a ping-pong protocol so that operators can operate on half words (thus introducing commutativity of the operators) causes the security problem to become undecidable.

1. INTRODUCTION

The use of public-key encryption [DH, RSA] for the secure network communication has received considerable attention. Such systems are effective against a "passive" eavesdropper, namely, one who merely taps the communication line and tries to decipher the intercepted message. However, as pointed out by Needham and Schroeder [NS], an improperly designed protocol can be vulnerable to "active" sabotage.

The saboteur may be a legitimate user in the network. He can intercept and alter messages, impersonate other users or initiate instances of the protocol between himself and other users, in order to use their responses. It is possible that through such complex manipulations he can read messages, which are supposed to be protected, without cracking the cryptographic systems in use.

In view of this danger it is desirable to have a formal model for discussing security issues in a precise manner, and to investigate the existence of efficient algorithms for checking the security of protocols.

Dolev and Yao [DY] investigated the security of two-person, ping-pong protocols; this investigation, was carried on by Dolev, Even and Karp [DEK]. In Section 2, we formulate the notion of multi-party ping-pong protocols and the related security problem. It was shown in DEK that if a two-party protocol is insecure, one saboteur suffices to demonstrate it and hence only one saboteur need be considered for checking the security of a two-party ping-pong protocol. In Section 3 we show that $3(p-2)+1$ and $3(p-2)+2$ are lower and upper bounds on the number of saboteurs which should be considered for the security problem of a p-party ping-pong protocol and conclude that for a fixed p these protocols can be tested for security in $O(n^3)$ time and $O(n^2)$ space, when n is the length of the protocol.

In Section 4 we show that the security problem is NP-Hard if the number of participants in the protocol is part of its input. In Section 5 we relax the definition of a ping-pong protocol and get a class of two-party protocols for which the security problem is undecidable. In Section 6 we discuss the problem of finding the shortest insecurity string and the power of name appending.

2. MULTI-PARTY PING-PONG PROTOCOLS AND THEIR SECURITY

Let N be the set of members in a communication network and Σ be a set of operators. Some operators may have a user name subscript (hereafter called index). We assume that Σ consists of operators such as encryption and decryption of Public-Key Cryptosystems (PKCS), name appending and deletion [NS, DY, DEK], and other functions. Note that there may be several PKCS instances per user and the same applies for the name appending/deletion mechanisms and other functions.

We will denote instances of x's PKCS encryption and decryption by $E_x^{(j)}$ and $D_x^{(j)}$ respectively. The name appendings/deletions of user x will be denoted by $i_x^{(j)}$ and $d_x^{(j)}$ respectively. In case there is only one PKCS instance [name appending/deletion mechanism] per user we will take the liberty of omitting the superscript (j). The subset of operators, which user x can perform is denoted by Σ_x and is called x's vocabulary. We assume that $\Sigma_x = \Sigma - \{D_y^{(j)}: y \in N-\{x\}, 1 \leqslant j \leqslant q\}$, where q is the number of PKCS instances per user. Note that the vocabularies of all users are similar in the sense that if one replaces the index x by y, and y by x, in Σ_x, the result is Σ_y.

Also, there is a given set of cancellation rules of the form $\sigma\tau \equiv \lambda$[+], where $\sigma,\tau \in \Sigma$. If both are indexed then the indices are the same. The cancellation rules are similar for all users. Thus, if one or both operators are indexed then the same cancellation rule holds for every index. Note that it may be the case that both $\sigma\tau \equiv \lambda$ and $\tau\sigma \equiv \lambda$ hold. In this case, we say that the cancellation rule $\tau\sigma \equiv \lambda$ is unordered. Note that the cancellation rules of the name appending/deletion mechanism (i.e. $d_x^{(j)} \cdot i_x^{(j)} \equiv \lambda$) is ordered. We will denote by $i_x^{(j)}$, $d_x^{(j)}$ all pairs of operators which are not secret and are ordered.

Note that if $a,b,c \in \Sigma$, $ab \equiv \lambda$ and $bc \equiv \lambda$ then $a \equiv c$. This follows from the fact that members of Σ are operators: Let $w \in \{0,1\}^*$.

$abc(w) = a(bc(w)) = a(w)$, since $bc \equiv \lambda$, but on the other hand,

$abc(w) = ab(c(w)) = c(w)$.

Thus $a \equiv c$.

Given a string $\alpha \in \Sigma^*$, one may repeatedly apply cancellation rules until no cancellation rule is applicable any more. By the previous paragraph, the reduction process has the Church Rosser property [R], and thus the end result, hereafter called the reduced form of α, is unique. Let us denote the reduced form of α by $\bar{\alpha}$.

[+] We say that two sequences of operators, α and β, are equal if they are equal as words over Σ, and denote this by $\alpha = \beta$. We say that two sequences of operators, α and β, are equivalent if for every $w \in \{0,1\}^*$ applying α to w and applying β to w yields the same result (i.e. $\alpha(w) = \beta(w)$), and denote this by $\alpha \equiv \beta$.

(Note that $\alpha \equiv \bar{\alpha}$ and that $\alpha \equiv \beta$ if $\bar{\alpha} = \bar{\beta}$.)

An underlying assumption in our analysis is that the set Σ is free from any relations other than those implied by cancellation rules. Thus, two strings of operators, α and β are equivalent if and only if both have the same reduced form.

Examples and demonstrations can be found in DEK.

We will assume throughout this work that $p \geqslant 2$.

We define a p-party Ping-Pong Protocol as a sequence of operator-words $\{\alpha_j(\underline{x})\}_{j=1}^{\ell}$ where $\underline{x} = (x_1, x_2, \ldots, x_p)$ is a sequence of user-name-variables, and for every j there exists an i_j, $1 \leqslant i_j \leqslant p$, such that $\alpha_j(\underline{x}) \in (\Sigma_{x_{i_j}})^*$; j specifies the phase in which the operator-word $\alpha_j(\underline{x})$ is applied by user x_{i_j}. Sometimes we will write α_j instead of $\alpha_j(\underline{x})$, provided $\underline{x}$ is defined before. We shall always assume that $\alpha_j(\underline{x}) = \overline{\alpha_j(\underline{x})}$; however, all the results hold even if this assumption is not made.

An $\underline{a}$-instance of a p-party ping-pong protocol is an assignment of the participants $\underline{a} = (a_1, \ldots, a_p)$ to the variable-users $\underline{x}$. In the j-th phase of the $\underline{a}$-instance of protocol P, hereby denoted by $P(\underline{a})$, a_{i_j} applies $\alpha_j(\underline{a})$, which denotes the $\underline{a}$-instance protocol word obtained by assigning the participants $\underline{a}$ to the variable-users $\underline{x}$. This operator word is applied (by a_{i_j}) to the message transmitted in the (j-1)-st phase if $j > 1$; for $j = 1$ it is applied to some message, M, initiated by a_{i_1}. In either case, a_{i_j} transmits the result with a statement specifying the name of the protocol ("P"), the phase (j) and the actual participants ($\underline{a}$).

Let us denote by A the set of users in $\underline{a}$. We will assume that $|A| = p$ (i.e. that one user is not allowed to play the role of different variable-users in an instance of a protocol) and that an honest user would refuse to participate in an instance of a protocol which does not meet this condition (suspecting that this instance is used for some illegitimate purpose). For further discussion of this assumption and what happens if it is not made, see Appendix A.

A protocol, P, is said to be insecure if there exists an instance of it, P($\underline{a}$), such that a user, s, not in A can, perhaps via collaboration with other users in the net, get the original message M through a fixed, predetermined, sequence of actions. The user s, and all the users not in A, who collaborate with him knowingly (see action (2) below) or unknowingly (action (3)), are called saboteurs. It is assumed that users in A do not collaborate with s knowingly but they may help him inadvertently, by participating in other instances of the protocol. Due to the assumption of freeness, the assumption that the saboteurs' sequence of actions is fixed and predetermined, is no weakening of the saboteurs' options.

Actions which s can take are:

(1) Obtain any message transmitted openly in the $\underline{a}$-instance of the protocol. Note that this is a purely "passive" eavesdropping.

(2) Apply any operator of a saboteur's vocabulary to any message. Note that s cannot apply directly an operator which is in z's vocabulary, but not in that of s, unless z, knowingly, collaborates with him.

(3) Apply any $\underline{b}$-instance of a protocol word ($\alpha_j(\underline{b})$) to any message, where $B \subset (A \cup S)$ and $|B| = p$. This may be done by waiting for P($\underline{b}$) to occur (or convincing b_{i_1} to initiate it), replacing the (j-1)-th message by the desirable message (if $j > 1$;

otherwise convincing b_{i_1} to choose it as the initial message) and reading the j-th transmission.

When a party, b_{i_j}, reacts to a message by applying $\alpha_j(\underline{b})$ to it, according to the protocol, he has no reason to suspect that his action may help someone to illegitimately seize M.

Let us denote $\Sigma_S \triangleq \bigcup_{\zeta \in S} \Sigma_\zeta$. Note that Σ_S is the set of all operators s can apply to any message (i.e. action (2)). For protocol P and a set of users T, define $I(P,T) \triangleq \{\alpha_i(\underline{b}): 1 \leq i \leq \ell, B \subset T, |B| = p\}$. $I(P,S \cup A)$ is the set of all instances of protocol words, where the p users, in the instance, are a subset of $S \cup A$. Note that $I(P,S \cup A)$ constitutes action (3).

We define protocol P to be <u>insecure</u> iff there exists a set of saboteurs, S, and a γ such that $\gamma \in (\Sigma_S \cup I(P,S \cup A))^*$ and $\overline{\gamma \cdot \alpha_1(\underline{a})} = \lambda$. Note that the choice of $\underline{a}$ is immaterial. We call $\gamma \cdot \alpha_1(\underline{a})$ an <u>insecurity string</u> of P. Consider a parsing (partition) of γ into words in $\Sigma_S \cup I(P,S \cup A)$ i.e. $\gamma = \gamma_n \cdots \gamma_2 \cdot \gamma_1$ where $\gamma_i \in \Sigma_S \cup I(P,S \cup A)$, for $1 \leq i \leq n$. We call γ_i a <u>filler</u> if $\gamma_i \in \Sigma_S$; otherwise γ_i is a word of $I(P,S \cup A)$. Throughout the paper, when we talk about an insecurity string [or about some $\gamma \in (\Sigma_S \cup I(P,S \cup A))^*$], we assume some fixed parsing of it. Consider a reduction process, which reduces $\gamma \cdot \alpha_1(\underline{a})$ to λ. Throughout the paper, when we talk about an insecurity string, we assume some reduction process on it.

Let θ be an occurrence of some operator in the insecurity string, we call the occurrence of the operator which cancels it, in the reduction process, θ's <u>mate</u>.

[The notion of insecurity can be extended for an environment (set) of protocols as follows: An environment $\{P_i\}_{i=1}^{q}$, where $P_i \triangleq \{\alpha_j^i(\underline{x})\}_{j=1}^{\ell_i}$ is a p_i-party ping-pong protocol, is said to be insecure iff there exist a protocol P_r, a set of saboteurs S and a $\gamma \in (\Sigma_S \cup \{\alpha_j^i(\underline{b}) : 1 \leq j \leq \ell_i,\ B \subset (A \cup S),\ |B| = p_i\})^*$, such that $\gamma \cdot \alpha_1^r(\underline{a}) = \lambda$. The results of Sections 3, 4 and 5 can be extended to apply to the security problem of environments of protocols.]

In this work, when we introduce a protocol, we do not assume any purpose for which this protocol will be executed. Furthermore, we even do not assume that during a honest execution of it a party to it will be able to read M or even that the protocol can be executed honestly, (i.e. that the result of the application of the i-th protocol word to the $(i-1)$-st transmission is defined). If a protocol, we present, "does not make sense", it can be viewed as part of some other protocol or part of a collection (i.e. environment) of protocols.

3. BOUNDS ON THE NUMBER OF SABOTEURS

Having defined the notion of insecurity of a p-party ping-pong protocol we introduce the following problem, hereby referred to as the Minimum number of Saboteurs which demonstrate the Insecurity of a p-party Protocol problem (MSIP(p)):

What is the minimum cardinality of S such that for every insecure p-party ping-pong protocol, $P \triangleq \{\alpha_i\}_{i=1}^{\ell}$, there exist a γ such that $\gamma \in (\Sigma_S \cup I(P, S \cup A))^*$ and $\overline{\gamma \cdot \alpha_1(\underline{a})} = \lambda$.

Theorem 1: If Σ consists only of encryptions and decryptions of PKCS's (i.e. several encryption decryption mechanisms per user) then the solution of MSIP(p) is (p-1).

Proof: Note, first, that (p-1) saboteurs are needed to allow a saboteur to read the original message in an instance of the following insecure protocol:

$$\alpha_1 \triangleq E^{(0)}_{x_p} \quad , \quad \alpha_2 \triangleq E^{(1)}_{x_{p-1}} \cdots E^{(1)}_{x_1} \cdot D^{(0)}_{x_p} \quad .$$

We now prove that (p-1) saboteurs are always sufficient to demonstrate the insecurity of an insecure protocol over PKCS's. Let P be such an insecure protocol, S a set of saboteurs and γ be a string in $(\Sigma_S \cup I(P,S \cup A))^*$ such that the reduced form of $\gamma \cdot \alpha_1(\underline{a})$ is λ. Also, assume that among the possible strings, γ contains the minimum number of words of $I(P,S \cup A)$; and furthermore, that the number of fillers is minimal.

Clearly, no word of $I(P,S \cup A)$ contains decryptions of several users and all the non-decryption operators (i.e. the encryptions) can be performed by any user. Thus, γ contains only words of $I(P,S \cup A)$ which include decryptions by some user in A (note that a word of $I(P,S \cup A)$ which does not contain such a decryption can be replaced by the appropriate operators in Σ_S, i.e. by fillers). By the minimality assumptions on γ, if two operators of the saboteur cancel in the cancellation pattern of γ then one of the operators occurs in a word of $I(P,S \cup A)$ and the other is a filler. We replace, independently, in each word of $I(P,S \cup A)$ in γ, the saboteurs which occur in that word by saboteurs which belong to a global set of p-1 saboteurs, denoted S'.

This replacement changes the indices of operators which occur in words of $I(P,S \cup A)$ and are indexed by elements of S. To preserve the cancellation pattern we replace the indices of their mates accordingly.

Note that this last replacement affects only fillers and thus no repeated replacement occurs. A formal description of this process which transforms $\gamma \in (\Sigma_S \cup I(P,S \cup A))^*$ into $\gamma'' \in (\Sigma_{S'} \cup I(P,S' \cup A))^*$ such that $|S'| \leq p-1$ and $\overline{\gamma''\cdot\alpha_1(\underline{a})} = \lambda$ follows:

Replace each $\beta(\underline{b}) \in I(P,S \cup A)$, which appears in γ by the word $\beta(\underline{b}')$, where for $1 \leq i \leq p$ $b_i' = b_i$ if $b_i \in A$; otherwise b_i' is some element of S' such that for $i \neq j$, $b_i' \neq b_j'$. The string which results is denoted γ'.

Let $\theta_x \in \Sigma_S$ be a filler and θ_x' its mate. Let $\theta_{x'}'$ be the operator which replaces θ_x' in the process of obtaining γ'. We now replace θ_x by $\theta_{x'}$. Note that the reduction pattern which applies to $\gamma\cdot\alpha_1(\underline{a})$ still applies to $\gamma''\cdot\alpha_1(\underline{a})$.

Q.E.D.

We say that a string, of words over an infinite alphabet of variables, is <u>paired</u> if the occurrences of the variables are partitioned into pairs such that:

(1) The elements of a pair are occurrences of the same variable and occur in different words.

(2) There exist no two pairs such that one and only one of the elements of the first pair occurs in the string between the occurrences of the elements of the second pair (i.e. the pairs constitute a well-formed parentheses expression).

We say that a paired string is linked if there exists a route between any two occurrences of the same variable, where a route is defined recursively as follows:

(1) There is a route between two occurrences of the same variable in the same word.

(2) There is a route between two occurrences which are in the same pair.

(3) If there is a route between θ_1 and θ_2 and a route between θ_2 and θ_3 then there is a route between θ_1 and θ_3 (i.e. transitivity).

(Note that being paired is a precondition to a string being linked.)

We define the following word problem, hereafter referred to by String Assignment (SA):

Given a linked string find the minimum number of constants which can be assigned to the variables such that the same constant is not assigned to different variables, if there is a word of the string in which both occur.

Define a t-string to be a linked string in which every word consists of occurrences of at most t different variables (while a variable may appear several times in the same word).

Define the Symbol Assignment of t-Strings problem (SAS(t)) to be the following question: What is the maximum solution of the SA problem when restricted to t-strings?

Throughout this section assume $t > 1$ and $p > 2$.

Lemma 1: For every insecure p-party ping-pong protocol P which requires q saboteurs in order to demonstrate its insecurity, there exist a (p-1)-string whose SA's solution is equal to q.

Proof: Let $P \overset{\Delta}{=} \{\alpha_i(\underline{x})\}_{i=1}^{\ell}$ be an insecure p-party ping-pong protocol and q be an integer such that q saboteurs are necessary and sufficient to demonstrate P's insecurity. Let S be a set of q saboteurs and $\gamma \in (\Sigma_S \cup I(P,S \cup A))^*$ be a string which contains the minimum number of words from $I(P,S \cup A)$, such that $\overline{\gamma \cdot \alpha_1(\underline{a})} = \lambda$. Note that each non-filler contains at least one non-saboteur decryption and thus the number of saboteurs which occur in words of γ does not exceed (p-1). Also note that omitting a non-saboteur operator and its mate does not relax the constraints on the number of saboteurs in γ, since these constraints are embedded in the relations among saboteurs' operators.

We say that two occurrences of operators indexed r are related if there exist a path between them, when a path is defined recursively as follows:

(1) There is a path between two operators which occur in the same word and are indexed by the same user.

(2) There is a path between two operators which cancel each other (i.e. mates).

(3) If there is a path between θ_1 and θ_2 and a path between θ_2 and θ_3 then there is a path between θ_1 and θ_3.

(Note that the definition of a path between occurrences of operators in γ is similar to the definition of a route between occurrences of a variable in a paired string.)

We present the following (p-1)-string: Assign a different variable to each set of occurrences of operators indexed by a saboteur such that the occurrences are related. Replace, in γ, each

occurrence of an operator by the variable assigned to it. Non-saboteur operators are omitted. Define two occurrences of a variable as a pair if the occurrences of the operators which they have replaced have been mates. Note that the resulting string is indeed paired, linked and that at most (p-1) different variables occur in each of its words. Note that the solution to the SA problem for the resulting string is q.
(Note that the constants can be assigned to the string according to the way the saboteurs appeared in $\gamma\cdot\alpha_1(\underline{s})$. On the other hand, it is not possible to assign less than q constants to the string since one can use such an assignment to derive an insecurity string for P which is similar to $\gamma\cdot\alpha_1(\underline{s})$ but uses less than q saboteurs.) The lemma follows.

□

Lemma 2: For every t-string whose SA's solution is q, there exist an insecure $(t+1)$-party ping-pong protocol which requires q saboteurs to demonstrate its insecurity.

Proof: Let $w_\ell \cdots w_2\cdot w_1$ be a t-string of minimal length such that q is the solution of its SA problem. We shall introduce a $(t+1)$-party protocol which is insecure and cannot be cracked by less than q saboteurs.

In each w_j we replace the variables by operators indexed by user-variables $x_1,x_2,\ldots,x_t$. This replacement is performed consistently within each word (but independently of other words). Let us denote by $\varphi_j(x_k)$ the variable of w_j which is replaced by x_k.

The operator is a name-appending if the occurrence of the variable is the r.h.s. element in its pair; otherwise the operator is an indexed name-deletion. Denote by w_j' the word which results from w_j.

Define the following (t+1)-party ping-pong protocol (for user variable $x_1, x_2, \ldots, x_t, x_{t+1}$):

$$\alpha_1 \triangleq E_{x_{t+1}} \cdot i_{x_{t+1}}$$

$$\alpha_{j+1} \triangleq E_{x_1} \cdot E_{x_2} \cdots E_{x_t} \cdot E_{x_{t+1}} \cdot (i_{x_{t+1}})^{j+1} w_j' (d_{x_{t+1}})^{j} \cdot D_{x_{t+1}}, \qquad \text{for } 1 \leq j \leq \ell$$

$$\alpha_{\ell+2} \triangleq (d_{x_{t+1}})^{\ell+1} \cdot D_{x_{t+1}} .$$

First let us show that the protocol, hereafter denoted P, is insecure and that a set of q saboteurs, denoted S, suffices to demonstrate its insecurity. To this end we introduce the string $\beta_{\ell+2} \cdot \beta_{\ell+1} \cdots \beta_2 \cdot \beta_1$, where $\beta_j \in I(P, S \cup A) \cdot \Sigma_S^*$ for $1 \leq j \leq \ell+2$. Let $\beta_1 \triangleq \alpha_1(\underline{a})$, $\beta_{\ell+2} \triangleq \alpha_{\ell+2}(\underline{a})$ and $\beta_{j+1} \triangleq \alpha_{j+1}(\underline{b}^{(j)}) \cdot D_{b_t^{(j)}} \cdots D_{b_2^{(j)}} \cdot D_{b_1^{(j)}}$ for $1 \leq j \leq \ell$.

With no loss of generality, we can assume that the names of the saboteurs in S are the constants assigned to the variables of $w_\ell \cdots w_2 \cdot w_1$ in the solution of the SA problem. For $1 \leq i \leq t$ and $1 \leq j \leq \ell$, $b_k^{(j)}$ is the saboteur whose name has been assigned to the variable $\varphi_j(x_k)$. For $1 \leq j \leq \ell$, $b_{t+1}^{(j)} = a_{t+1}$.

Note that $\overline{\beta_{\ell+2} \cdot \beta_{\ell+1} \cdots \beta_2 \cdot \beta_1} = \overline{w_\ell' \cdots w_1'} = \lambda$. Thus, $\beta_{\ell+2} \cdot \beta_{\ell+1} \cdots \beta_2 \cdot \beta_1$, demonstrates P's insecurity using q saboteurs.

Having shown that P is insecure, Let S' be a set of saboteurs and $\gamma \in (\Sigma_{S'} \cup I(P,S' \cup A))^*$ such that $\overline{\gamma \cdot \alpha_1(\underline{a})} = \lambda$ and γ contains the minimum number of words of $I(P,S' \cup A)$. The following facts are of interest:

(1) In each instance of protocol word (i.e. a word of $I(P,S' \cup A)$) which occurs in γ the role of x_{t+1} is played by a_{t+1}. In each appearance of α_{j+1} in γ saboteurs are assigned to all the other x_i's, since there is no protocol word which can remove the string $E_{x_1} \cdot E_{x_2} \cdots E_{x_t}$ of α_{j+1}.

(2) $\overline{\alpha_{j+1}} = \alpha_{j+1}$, for $1 \leq j \leq \ell$. This follows from $\overline{w_j'} = w_j'$ and the fact that on the r.h.s. of w_j' appears a $d_{x_{t+1}}$ operator while on its l.h.s. appears an $i_{x_{t+1}}$ operator.

(3) The string γ contains an instance of $\alpha_{\ell+2}$.

(4) The j-th protocol word in γ (counting from right to left) is an instance of α_{j+1}.

(Using induction on j one can prove that the number of $d_{x_{t+1}}$ in the j-th protocol word is j. Note that the claim holds for $j=1$. Assume that the claim holds for every $k < j$. Note that the $d_{x_{t+1}}$'s which occur in the j-th protocol word must be canceled by the $i_{x_{t+1}}$'s which occur in the $(j-1)$-st protocol word. Also note that, by the minimality of $w_\ell \cdots w_2 \cdot w_1$, the $i_{x_{t+1}}$'s which occur in the $(j-1)$-st protocol word must be canceled by the $d_{x_{t+1}}$'s which occur in the j-th protocol word. Thus, the claim holds for j.)

(5) $\lambda = \overline{\gamma \cdot \alpha_1(\underline{a})} = \overline{w_\ell' \cdots w_2' \, w_1'}$.

(6) $|S'| \geqslant q$. Assuming on the contrary that $|S'| < q$ contradicts our assumption that the solution to $w_\ell \cdots w_2 w_1$'s SA problem is q (since the constants can be assigned to the w_j's according to the assignment of saboteurs to the w_j''s). □

The lemma remains valid even if MSIP(t+1) is restricted by one of the following:

(i) Σ consists of encryptions and decryptions by a single PKCS and a single name appending/deletion mechanism per user (i.e. $\Sigma = \{E_x, D_x, i_x, d_x \colon x \in N\}$).

(ii) The cancellation rules are unordered (i.e. if $\sigma\tau \equiv \lambda$ is a cancellation rule then so is $\tau\sigma \equiv \lambda$).

(Note that the proof of the lemma uses only protocols restricted by (i). To prove the validity of the lemma under restriction (ii) introduce the following protocol, for a given string $w_\ell \cdots w_2 \cdot w_1$:

$$\alpha_1 \triangleq E^{(2)}_{x_{t+1}} \cdot D^{(0)}_{x_{t+1}} , \quad \alpha_{\ell+2} \triangleq E^{(0)}_{x_{t+1}} \cdot D^{(\ell+2)}_{x_{t+1}} \quad \text{and}$$

$\alpha_{j+1} \triangleq E^{(1)}_{x_t} \cdots E^{(1)}_{x_1} \cdot E^{(j+2)}_{x_{t+1}} \cdot w_j'' \cdot D^{(j+1)}_{x_{t+1}}$ for $1 \leqslant j \leqslant \ell$. w_j'' is obtained from w_j by replacing, as in the proof of Lemma 2, its variables by operators indexed by user-variables $x_1, x_2, \ldots, x_t$. The operator is a name-dependent function if the occurrence of the variable is the r.h.s. element in its pair, otherwise the operator is the inverse of this function. We use a different function for every pair; it may be possible to reduce the number of functions used by considering the specific w_i's.)

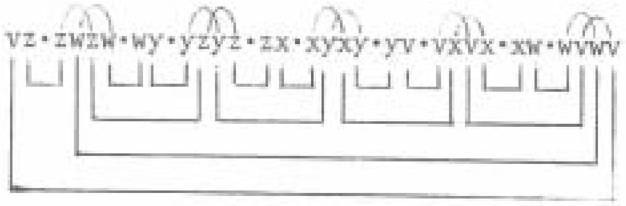
Corollary 1: If SAS(t) has a solution then it is equal to the solution of MSIP(t+1).

Proof: Follows from Lemmas 1 and 2.

□

Lemma 3: 5 is a lower bound on the solution of SAS(2).

Proof: Consider problem SA for the following 2-string (Fig.1):



The lines below the string show the partition into pairs while the lines above show routes within words.

Figure 1

It is easy to see that the solution to this instance of the SA problem is 5, since no two variables can be assigned the same constant.

□

Lemma 4: 3(t-1) + 1 is a lower bound on the solution of SAS(t).

Proof: Consider the following t-string (Fig. 2)

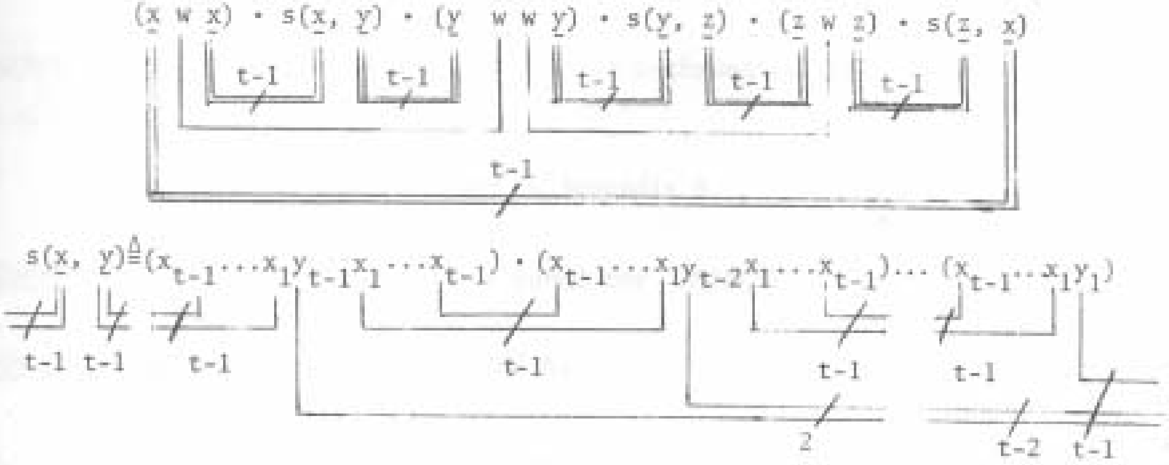

(x w x) · s(x, y) · (y w w y) · s(y, z) · (z w z) · s(z, x)
t-1
t-1
t-1
t-1
t-1
t-1
s(x, y) ≜ (x_{t-1}...x_1 y_{t-1} x_1 ... x_{t-1}) · (x_{t-1}...x_1 y_{t-2} x_1 ... x_{t-1}) ... (x_{t-1}...x_1 y_1)
t-1
t-1
t-1
t-1
t-1
t-1
2
t-2
t-1

Figure 2

It can be verified that no two variables can be assigned the same constant. Thus one should use $3(t-1)+1$ constants for this "instance" of SAS(t).

□

Theorem 2: 5 and $3(p-2)+1$ are lower bounds for MSIP(3) and MSIP(p), respectively.

Proof: Immediate by Lemmas 2, 3 and 4. Q.E.D.

□

Given a t-string, s, we define $G_s(V_s, E_s)$ to be the following undirected graph: V_s is the set of variables in s and E_s consists of pairs of variables such that there exists a word in the string s in which both variables occur. The following lemma is immediate:

Lemma 5: k is the solution of the SA problem when applied to a t-string s iff G_s is k-chromatic.

Notice that the graphs, of the strings presented in Lemmas 3 and 4 are cliques of 5 and $3(t-1)+1$ vertices respectively.

The following combinatorial lemma concerns the chromatic number of G_s, when s is a t-string.

Lemma 6: If s is a t-string and G_s is c-chromatic then $c \leq 3(t-1)+2$.

The proof of Lemma 6 is given in Appendix B.

Theorem 3: $3(p-2)+2$ is an upper bound for MSIP(p).

Proof: Immediate by Lemmas 1, 5 and 6.

Q.E.D.

Corollary 2: 5 is the solution to MSIP(3).

We conjecture that for $p > 3$, $3(p-2)+1$ is the solution to MSIP(p).

Corollary 3: For every fixed p, the security of a p-party ping-pong protocol can be tested in time $O(n^3)$ and space $O(n^2)$, when n is the length of the protocol.
This follows easily from technics in DEK.

4. ON THE NP-HARDNESS OF THE SECURITY PROBLEM WHEN THE NUMBER OF PARTICIPANTS IS PART OF ITS INPUT

As shown in the previous section, for every fixed p, the security of a p-party ping-pong protocol can be tested in polynomial time. In this section we show that this is unlikely to be the case for an unfixed p. To this end we formulate the following decision problem, hereafter referred to as the Security of Multi-Party Ping-Pong Protocol problem (SMPP): Given a multi-party ping-pong protocol determine whether it is insecure. We have found it convenient to reduce a restricted 3XC (R3XC) problem, defined below, to the *SMPP* problem.

R3XC problem: Given a set $U = \{e_i\}_{i=1}^{3n}$ and a collection of three-element subsets $S = \{s_j\}_{j=1}^{3n}$ of U, such that every element of U appears exactly in three subsets, determine whether there exists a subcollection of S such that every element of U appears exactly in one subset of the subcollection.

Lemma 7: The R3XC problem is NP-Complete.

The proof is given in Appendix C and is by a reduction from the 3XC problem.

Theorem 4: The SMPP problem is NP-Hard, even if $\Sigma = \{E_X, D_X, i_X, d_X : x \in N\}$ (i.e. consists of encryption and decryption of a single PKCS and a single name appending/deletion mechanism, per user).

Proof: By reduction from the R3XC problem. The idea of the reduction is to simulate in an insecurity string of the protocol (of the SMPP instance) a correct guess of the solution to the R3XC instance and its verification, provided such a solution exists.

Given an instance $U = \{e_i\}_{i=1}^{3n}$, $S = \{s_i\}_{i=1}^{3n}$ of R3XC, define $f_j(i)$ to be the index of the j-th subset which contains e_i (i.e. if $e_i \in s_{k_j}$ for $1 \leqslant j \leqslant 3$ and $k_1 < k_2 < k_3$ then $f_j(i) = k_j$ for $1 \leqslant j \leqslant 3$).

Given (U,S), we introduce the following (instance of SMPP which is a) (6n+1)-party ping-pong protocol, denoted $P(\underline{x})$, where $\underline{x} = (x_o, x_1, \ldots, x_{6n})$:

The first word of $P(\underline{x})$, $\alpha_1(\underline{x})$, is $E_{x_o} \cdot i_{x_o} \cdot I'(\underline{x}) \cdot B_1(\underline{x}) \cdot B_2(\underline{x}) \cdots B_{3n}(\underline{x}) \cdot i_{x_o}$, where $I'(\underline{x}) \triangleq i_{x_{3n}} \cdots i_{x_1} \cdot i_{x_o}$ and

$B_i \triangleq i_{x_{3n+f_1(i)}} \cdot i_{x_{3n+f_2(i)}} \cdot i_{x_{3n+f_3(i)}}$, for $1 \leqslant i \leqslant 3n$.

[Note that $\alpha_1(\underline{x})$ fully encodes the structure of (U,S).]

Denote $C'(\underline{x}) \triangleq d_{x_o} \cdot d_{x_1} \cdots d_{x_{3n}}$, $I(\underline{x}) \triangleq i_{x_{4n}} \cdots i_{x_{3n+2}} \cdot i_{x_{3n+1}} \cdot I'(\underline{x})$,
and $C(\underline{x}) \triangleq C'(\underline{x}) \cdot d_{x_{3n+1}} \cdot d_{x_{3n+2}} \cdots d_{x_{4n}}$.

$\alpha_2(\underline{x}) \triangleq E_{x_1} \cdot I(\underline{x}) \cdot C'(\underline{x}) \cdot d_{x_o} \cdot D_{x_o}$. [Note that the correspondance between (the instances) $\alpha_1(\underline{a})$ and $\alpha_2(\underline{b})$ relates to a subcollection, of cardinality n, of S, provided that $\{b_i\}_{i=3n+1}^{4n} \subset \{a_i\}_{i=3n+1}^{6n}$. Such a subcollection may be an exact cover of U, i.e. it is a guess, which may be correct, of an exact cover of U.]

Denote $C_{1,k}(\underline{x}) \triangleq d_{x_{4n+2}} \cdot d_{x_{4n+1}} \cdot d_{x_{3n+k}}$,

$C_{2,k}(\underline{x}) \triangleq d_{x_{4n+2}} \cdot d_{x_{3n+k}} \cdot d_{x_{4n+1}}$, and

$C_{3,k}(\underline{x}) \triangleq d_{x_{3n+k}} \cdot d_{x_{4n+2}} \cdot d_{x_{4n+1}}$, for $1 \leqslant k \leqslant n$.

$\alpha_{i,j,k}(\underline{x}) \triangleq E_{x_{i+1}} \cdot I(\underline{x}) \cdot C_{j,k}(\underline{x}) \cdot C(\underline{x}) \cdot D_{x_i}$, for $1 \leqslant i \leqslant 3n$, $1 \leqslant j \leqslant 3$ and $1 \leqslant k \leqslant n$.

[Note that $\overline{C_{j,k}(\underline{b}') \cdot B_i(\underline{a})} = \lambda$ only if $\{b'_{3n+k}, b'_{4n+1}, b'_{4n+2}\} = \{a_{f_q(i)} : 1 \leq q \leq 3\}$. This fact is used to verify that there is a (unique) subset in the subcollection, introduced by the correspondence between $\alpha_1(\underline{a})$ and $\alpha_2(\underline{b})$, which covers e_i.]

$\alpha_3(\underline{x}) \triangleq d_{x_o} \cdot C(\underline{x}) \cdot D_{x_{3n+1}}$. [Note that $\alpha_3(\underline{x})$ is the only protocol word which contains a decryption and does not contain an encryption.]

$$P(\underline{x}) \triangleq \{\alpha_j(\underline{x}) : 1 \leqslant j \leqslant 3\} \cup \{\alpha_{i,j,k}(\underline{x}) : 1 \leqslant i \leqslant 3n,\ 1 \leqslant j \leqslant 3,\ 1 \leqslant k \leqslant n\}.$$

Let us prove that the reduction is valid. First, assume that there is an exact cover, denoted $C \triangleq \{s_{i_j}\}_{j=1}^{n}$, of U. Denote by k_q the index in C of the subset which contains e_q (i.e. if $e_q \in s_{i_j}$ then $k_q = j$). Denote by m_q the index in $\{s_{f_j(q)}\}_{j=1}^{3}$ of this subset (i.e. if $i_{k_q} = f_j(q)$ then $m_q = j$). Extend $\{i_j\}_{j=1}^{n}$ so that $\{i_j\}_{j=1}^{3n} = \{j\}_{j=1}^{3n}$. Let $\underline{a} \triangleq (a_o, a_1, \ldots, a_{6n})$ and $\underline{a}' \triangleq (a_o, a_1, \ldots, a_{3n}, a_{3n+i_1}, a_{3n+i_2}, \ldots, a_{3n+i_{3n}})$. Let $\underline{a}'(q) \triangleq (b_o, b_1, \ldots, b_{6n})$ such that $b_j = a_j$ for $0 \leqslant j \leqslant 4n$, $b_{4n+1} = a_{3n+f_r(q)}$ and $b_{4n+2} = a_{3n+f_t(q)}$, where $r < t$ and $\{m_q, r, t\} = \{1,2,3\}$. Note that $C_{m_q,k_q}(\underline{a}(q)) = d_{a_{3n+f_3(q)}} \cdot d_{a_{3n+f_2(q)}} \cdot d_{a_{3n+f_3(q)}}$, for $1 \leqslant q \leqslant 3n$. Note that $\alpha_3(\underline{a}') \cdot \alpha_{3n,m_{3n}k_{3n}}(\underline{a}'(3n)) \cdots \alpha_{2,m_2,k_2}(\underline{a}'(2)) \cdot \alpha_{1,m_1,k_1}(\underline{a}'(1)) \cdot \alpha_2(a') \cdot \alpha_1(\underline{a})$ is an insecurity string of P. Thus P is insecure.

Assume, on the other hand, that P is insecure. Assume, with no loss of generality, that the original participants of P are $\underline{a} = (a_o, a_1, \ldots, a_{6n})$. Let S' be a set of saboteurs and let $\gamma \in ((\bigcup_{\zeta \in S'} \Sigma_\zeta) \cup I(P, S' \cup A))^*$ have the minimum number of words such that $\overline{\gamma \cdot \alpha_1(\underline{a})} = \lambda$. Let $\gamma = w_\ell \cdots w_2 \cdot w_1$. The following facts about γ are of interest:

<u>Fact 1</u>: $w_1 = \alpha_2(\underline{b}^{(0)})$, where $\underline{b}^{(0)} \triangleq (b_o^{(0)}, b_1^{(0)}, \ldots, b_{6n}^{(0)})$ and $B^{(0)} \subseteq S' \cup A$. (Since $\alpha_2(\underline{x})$ is the only protocol word which can cancel $E_{a_o} \cdot i_{a_o}$.)

<u>Fact 2</u>: For $0 \leqslant q \leqslant 3n$, $b_q^{(0)} = a_q$. (Since $\overline{C'(\underline{b}^{(0)}) \cdot I'(\underline{a})} = \alpha$.)

<u>Fact 3</u>: For $1 \leqslant i \leqslant 3n$, $w_{i+1} = \alpha_{i,r_i,t_i}(\underline{b}^{(i)})$, where $B^{(i)} \subseteq S' \cup A$,

$1 \leqslant r_i \leqslant 3$ and $1 \leqslant t_i \leqslant n$. Furthermore, $b_q^{(i)} = b_q^{(i-1)} = b_q^{(0)}$, for $0 \leqslant q \leqslant 4n$. (Using induction on i, note that $\{\alpha_{i,j,k}(\underline{x}) : 1 \leqslant j \leqslant 3, 1 \leqslant k \leqslant n\}$ is the set of protocol words which can cancel $E_{a_i} \cdot I(b^{(i-1)})$ and that $E_{a_i} \cdot I(b^{(i-1)})$ appears in $\overline{w_i \cdots w_2 \cdot w_1 \cdot \alpha_1(a)}$. Also, $\overline{C(b^{(i)}) \cdot I(b^{(i-1)})} = \lambda$ and $b_q^{(i)} = b_q^{(i-1)}$, for $0 \leqslant q \leqslant 4n$, follows.)

Fact 4: For $1 \leqslant i \leqslant 3n$, $b_{3n+t_i}^{(i)} = a_{3n+f_{r_i}(i)}$. (Note that $\overline{C_{r_i,t_i}(\underline{b}^{(i)}) \cdot B_i(\underline{a})} = \lambda$ and $b_{3n+t_i}^{(i)} = a_{3n+f_{r_i}(i)}$ follows.)

[Fact 5: $w_{3n+2} = \alpha_3(b^{(3n+1)})$, $\ell = 3n+2$ and $b_q^{(3n+1)} = b_q^{(0)}$, for $0 \leqslant q \leqslant 4n$.]

Fact 6: For every $1 \leqslant i \leqslant 3n$, $e_i \in s_{f_{r_i}}(i)$. (By $f_j(i)$'s definition.)

Fact 7: If $t_i = t_j$ then $f_{r_i}(i) = f_{r_j}(j)$. (Combining Facts 3 and 4, we get $b_{3n+t_i}^{(0)} = b_{3n+t_i}^{(i)} = a_{3n+f_{r_i}(i)}$ and $b_{3n+t_j}^{(0)} = b_{3n+t_j}^{(j)} = a_{3n+f_{r_j}(j)}$. Assuming $t_i = t_j$, $a_{3n+f_{r_i}(i)} = a_{3n+f_{r_j}(j)}$ and $f_{r_i}(i) = f_{r_j}(j)$ follow.)

Fact 8: $|\{f_{r_i}(i) : 1 \leqslant i \leqslant 3n\}| \leqslant n$. (By Fact 7.)

Define $C \stackrel{\Delta}{=} \{s_{f_{r_i}(i)} : 1 \leqslant i \leqslant 3n\}$. By Fact 6 C covers U while by Fact 8 $|C| \leqslant n$. Thus, C is an exact cover of U.

Q.E.D.

We do not know whether the SMPP problem is in NP. We have reasons (see Appendix D) to believe that this is not the case. In fact, we conjecture that SMPP is neither in NP nor in Co-NP. However, using techniques which are presented in DEK, SMPP can be solved in exponential time (and exponential space).

5. ON THE UNDECIDABILITY OF THE SECURITY PROBLEM OF HALF-WORD PING-PONG PROTOCOLS

In the previous sections we were concerned with protocols over operators which are free of any relations other than those implied by the cancellation rules. We now relax this assumption allowing commutativity of some pairs of operators.

We define a Half-Word Ping-Pong Protocol to be a two-party ping-pong protocol over the following structure:

(1) $\Sigma = \{E^\pi_x, D^\pi_x : x \in N, \pi \in \{L,R,W\}\} \cup \{i^\pi_x, d^\pi_x : x \in N, \pi \in \{L,R\}\}$.

with the cancellation rules

$$E^\pi_x \cdot D^\pi_x \equiv \lambda \quad D^\pi_x \cdot E^\pi_x \equiv \lambda \qquad \text{for } x \in N, \pi \in \{L,R,W\}$$

and

$$d^\pi_x \cdot i^\pi_x \equiv \lambda \qquad \text{for } x \in N, \pi \in \{L,R\}.$$

(2) The semantics

For $\theta \in \{E,D\}$ and $x \in N$, θ^W_x is the extension of θ_x, of the previous sections (which operates on word from $\{0,1\}^*$), to operate on words over $\{0,1,\$\}$. ($E_x$ [D_x] is the encryption [decryption] of x's instance of a PKCS and i_x [d_x] is the appending [deletion] of his name.) For $\theta \in \{E,D,i,d\}$, $x \in N$, $\pi \in \{L,R\}$ and $w \in \{0,1,\$\}^*$, if $w \notin \{0,1\}^* \cdot \$ \cdot \{0,1\}^*$ then $\theta^\pi_x(w)$ is not defined; otherwise w can be written as $w_1\$w_2$ where $w_1, w_2 \in \{0,1\}^*$ and $\theta^L_x(w_1\$w_2) \triangleq \theta_x(w_1)\$w_2$, $\theta^R_x(w_1\$w_2) \triangleq w_1\$\theta_x(w_2)$.

(3) There are no relations among the operators other than those implied by the cancellation rules and the semantics.

We define the Security of Half-Word Ping-Pong Protocol problem (SHWP) as follows:

Given a half-word ping-pong protocol determine whether it is insecure. (The notion of insecurity of a Half-Word protocols is similar to the notion given in Section 2.)

Theorem 5: The SHWP problem is undecidable.

Proof: The Post-Correspondence Problem (PCP) was defined and shown to be undecidable by Post [P]. Its instance consists of two list, $Y \overset{\Delta}{=} \{y_i\}_{i=1}^n$ and $Z \overset{\Delta}{=} \{z_i\}_{i=1}^n$, of words over $\{0,1\}^*$ and one is asked to determine whether there exists a non-empty sequence of indices $i_1, i_2, \ldots, i_\ell$ such that $y_{i_\ell} \cdots y_{i_1} = z_{i_\ell} \cdots z_{i_1}$ (i.e. the strings are bit-wise equal).

We prove that the SHWP problem is undecidable by reducing the PCP problem to it. The idea of the reduction is to simulate in an insecurity string (of the SHWP instance) a correct guess of a solution to the PCP instance and its verification, provided such a solution exists. In the simulation process, we first construct a string of Y's words and a string of Z's words both corresponding to one, non-deterministically chosen, sequence of indices. Next we check deterministically, whether these strings are bit-wise equal.

Given an instance, $Y \overset{\Delta}{=} \{y_i\}_{i=1}^n$ and $Z \overset{\Delta}{=} \{z_i\}_{i=1}^n$, of the PCP problem, introduce the following half-word ping-pong protocol, denoted $P(\underline{x})$ where $\underline{x} = (x_o, x_1)$:
The first protocol word, $\alpha_1(\underline{x})$, is $E^W_{x_o} \cdot S^L(\underline{x}) \cdot S^R(\underline{x})$, where $S^\pi(\underline{x}) \overset{\Delta}{=} i^\pi_{x_o} \cdot i^\pi_{x_1} \cdot E^\pi_{x_o} \cdot i^\pi_{x_o}$ for $\pi \in \{L,R\}$.

Before presenting the next 2n protocol words (which fully encode Y and Z) we introduce the following denotations:

$I^{\pi}_{\sigma_m \cdots \sigma_2 \cdot \sigma_1}(\underset{\sim}{x}) \triangleq i^{\pi}_{x_{\sigma_m}} \cdots i^{\pi}_{x_{\sigma_2}} \cdot i^{\pi}_{x_{\sigma_1}}$ where $\pi \in \{L,R\}$ and $\sigma_q \in \{0,1\}$ for $1 \leq q \leq m$. $I^{\pi}_{j,k}(\underset{\sim}{x}) \triangleq i^{\pi}_{x_k} \cdot i^{\pi}_{x_{k \oplus 1}} \cdot I^{\pi}_{\tau}(\underset{\sim}{x}) \cdot d^{\pi}_{x_1} \cdot d^{\pi}_{x_o}$, where $\oplus$ denotes addition modulo 2, $\tau \triangleq y_j$ if $\pi = L$ and $\tau \triangleq z_j$

[Note that $I^{L}_{j,k}(\underset{\sim}{x})$ encodes y_j while $I^{R}_{j,k}$ encodes z_j.]

$\alpha_{2j+k}(\underset{\sim}{x}) \triangleq E^{W}_{x_o} \cdot I^{L}_{j,k}(\underset{\sim}{x}) \cdot I^{R}_{j,k}(\underset{\sim}{x}) \cdot D^{W}_{x_o}$, for $1 \leq j \leq n$ and $0 \leq k \leq 1$.

[These words will be used to simulate the guess of a string of Y's words and a string of Z's words which correspond to one sequence of indices. The simulation is by construction of a sequence of name-appending operators superscripted by L [R] which corresponds to the string of Y's [Z's] words. Only the last protocol word used for this purpose has an odd serial number.]

The next two protocol words, $\alpha_{2n+2}(\underset{\sim}{x})$ and $\alpha_{2n+3}(\underset{\sim}{x})$, will be used to check whether the strings, simulated by the above defined 2n words, are bit-wise equal.

$\alpha_{2n+2+\sigma}(\underset{\sim}{x}) \triangleq E^{W}_{x_o} \cdot C^{L}_{\sigma}(\underset{\sim}{x}) \cdot C^{R}_{\sigma}(\underset{\sim}{x}) \cdot D^{W}_{x_o}$ for $\sigma \in \{0,1\}$, where

$C^{\pi}_{\sigma}(\underset{\sim}{x}) \triangleq i^{\pi}_{x_1} \cdot i^{\pi}_{x_o} \cdot d^{\pi}_{x_\sigma} \cdot d^{\pi}_{x_o} \cdot d^{\pi}_{x_1}$, for $\pi \in \{L,R\}$.

$\alpha_{2n+4}(\underset{\sim}{x}) \triangleq F^{L}(\underset{\sim}{x}) \cdot F^{R}(\underset{\sim}{x}) \cdot D^{W}_{x_o}$, where $F^{\pi}(\underset{\sim}{x}) \triangleq d^{\pi}_{x_o} \cdot D^{\pi}_{x_o} \cdot d^{\pi}_{x_o} \cdot d^{\pi}_{x_1}$ for $\pi \in \{L,R\}$. Note that $P(\underset{\sim}{x}) \triangleq \{\alpha_j(x) : 1 \leq j \leq 2n+4\}$.

Let us prove that the reduction is valid. First, assume that $i_1, i_2, \ldots, i_\ell$ is a sequence of indices which constitutes a solution to the PCP instance (i.e. $\delta \triangleq y_{i_\ell} \cdots y_{i_2} \cdot y_{i_1} = z_{i_\ell} \cdots z_{i_2} \cdot z_{i_1}$).

Let a_o and a_1 be two users and $\underset{\sim}{a} \triangleq (a_o, a_1)$. Define $\beta_j \triangleq \alpha_{2 \cdot i_j}(\underset{\sim}{a})$, for $1 \leqslant j < \ell$, and $\beta_\ell \overset{\Delta}{=} \alpha_{2 \cdot i_\ell + 1}(\underline{a})$. Let $\delta = \sigma_1 \cdot \sigma_2 \cdots \sigma_m$, where $\sigma_q \in \{0,1\}$ for $1 \leqslant q \leqslant m$. Define $\beta_{\ell+q} \overset{\Delta}{=} \alpha_{2n+2+\sigma_q}(\underline{a})$, for $1 \leqslant q \leqslant m$, and $\beta_{\ell+m+1} \overset{\Delta}{=} \alpha_{2n+4}(\underline{a})$. Note that $\overline{\beta_{\ell+m+1} \cdot \beta_{\ell+m} \cdots \beta_2 \cdot \beta_1 \cdot \alpha_1(\underline{a})} = \lambda$. Thus, the SHWP instance ,$P(\underline{x})$, is insecure.

Assume, on the other hand, that $P(\underline{x})$ is insecure. Assume, with no loss of generality, that a_o and a_1 are the original users of the protocol, $\overline{\gamma \cdot \alpha_1(a_o, a_1)} = \lambda$ and $\gamma = w_q \cdots w_2 \cdot w_1$ has the minimum number of words (either instances of protocol words or fillers). The following facts about γ are of interest:

Denote $\beta'_j \overset{\Delta}{=} \overline{w_j \cdots w_2 \cdot w_1 \cdot \alpha_1(a_o, a_1)}$, for $0 \leqslant j < q$.

Fact 1: If there is no instance of $\alpha_{2n+4}(\underline{x})$ in the sequence $w_1, w_2, \ldots, w_j$, then $\{w_k : 1 \leqslant k \leqslant j\} \subseteq \{\alpha_k(a_o, a_1) : 2 \leqslant k < 2n+4\}$, the left-most operator of β'_j is $E^W_{a_o}$ and the two left-most operators of it which have superscript L [R], hereafter denoted $\mathrm{suf}^L(j)$ [$\mathrm{suf}^R(j)$], are either $i^L_{a_o} \cdot i^L_{a_1}$ [$i^R_{a_o} \cdot i^R_{a_1}$] or $i^L_{a_1} \cdot i^L_{a_o}$ [$i^R_{a_1} \cdot i^R_{a_o}$]. (Using induction on j notice that if the claim holds for j-1 then the right-most operator of w_j must be $D^W_{a_o}$. Thus, $w_j \in \{\alpha_k(a_o, y) : 2 \leqslant k < 2n+4,\ y \in N\}$. Also note that the two right-most operators of w_j which have superscript L must cancel $\mathrm{suf}^L(j-1)$. Thus, $w_j \in \{\alpha_k(a_o, a_1) : 2 \leqslant k < 2n+4\}$ and the claim holds for j. Vacuously, the claim holds for $j = 0$.)

Fact 2: If w_j is an instance of $\alpha_{2n+4}(\underline{x})$ then $j = q$ and $w_j = \alpha_{2n+4}(a_o, a_1)$.

(Denote by k the smallest integer such that w_k is an instance of α_{2n+4}. Using Fact 1 one can show that $w_k = \alpha_{2n+4}(a_o, a_1)$.

Using induction on $0 \leq j < k$ one can prove that the two right-most operators in β'_j which have superscript L [R] are $E^L_{a_o} \cdot i^L_{a_o}$ $[E^R_{a_o} \cdot i^R_{a_o}]$ and that $E^L_{a_o}$ $[E^R_{a_o}]$ appears only once in β'_j. Note that $D^L_{a_o}$ $[D^R_{a_o}]$ which appears in $\alpha_{2n+4}(a_o,a_1)$ must be cancelled by an operator of β'_{k-1}, since $d^L_{a_o}$ $[d^R_{a_o}]$ is on its l.h.s. Therefore, $\overline{\alpha_{2n+4}(a_o,a_1) \cdot \beta'_{k-1}} = \lambda$. By the minimality of γ, $k = q$ follows.)

Fact 3: $w_j \in \{\alpha_k(a_o,a_1) : 2 \leq k \leq 2n+3\}$, for $1 \leq j < q$, and $w_q = \alpha_{2n+4}(a_o,a_1)$. (By Facts 1 and 2.)

Fact 4: For $1 \leq j \leq q$, $\mathrm{suf}^\pi(j-1) = i^\pi_{a_1} \cdot i^\pi_{a_o}$ iff $w_j \in \{\alpha_k(a_o,a_1) : 2n+2 \leq k \leq 2n+4\}$. (Note that the two right-most operator of $w_j \triangleq \alpha_k(a_o,a_1)$ which have superscript π are $d^\pi_{a_o} \cdot d^\pi_{a_1}$ if $2n+2 \leq k \leq 2n+4$ and $d^\pi_{a_1} \cdot d^\pi_{a_o}$ if $2 \leq k \leq 2n+1$. Note that these operators must cancel $\mathrm{suf}^\pi(j-1)$ and therefore $2n+2 \leq k \leq 2n+4$ iff $\mathrm{suf}^\pi(j-1) = i^\pi_{a_1} \cdot i^\pi_{a_o}$.)

Fact 5: There is a unique r such that $1 \leq r < q$ and $w_r \in \{\alpha_{2k+1}(a_o,a_1) : 1 \leq k \leq n\}$. Furthermore, $w_j \in \{\alpha_{2k}(a_o,a_1) : 1 \leq k \leq n\}$ iff $j < r$. (Note that $\mathrm{suf}^\pi(0) = i^\pi_{a_o} \cdot i^\pi_{a_1}$ and $\mathrm{suf}^\pi(q-1) = i^\pi_{a_1} \cdot i^\pi_{a_o}$.
Also,note that $\mathrm{suf}^\pi(j-1) = i^\pi_{a_o} \cdot i^\pi_{a_1}$ and $\mathrm{suf}^\pi(j) = i^\pi_{a_1} \cdot i^\pi_{a_o}$ iff $w_j \in \{\alpha_{2k+1}(a_o,a_1) : 1 \leq k \leq n\}$. Furthermore, if $\mathrm{suf}^\pi(j-1) = \mathrm{suf}^\pi(j) = i^\pi_{a_o} \cdot i^\pi_{a_1}$ then $w_j \in \{\alpha_{2k}(a_o,a_1) : 1 \leq k \leq n\}$. By fact 4 if $\mathrm{suf}^\pi(j-1) = i^\pi_{a_1} \cdot i^\pi_{a_o}$ then $w_j \in \{\alpha_k(a_o,a_1) : 2n+2 \leq k \leq 2n+4\}$. Also note that if $w_j \in \{\alpha_k(a_o,a_1) : 2n+2 \leq k \leq 2n+3\}$ then $\mathrm{suf}^\pi(j) = i^\pi_{a_1} \cdot i^\pi_{a_o}$. Thus, Fact 5 follows.)

Assume, with no loss of generality, that $w_j = \alpha_{2\cdot i_j}(a_o,a_1)$ for $1 \leq j < r$ and $w_r = \alpha_{2\cdot i_r+1}(a_o,a_1)$.

Fact 6: $y_{i_r} \cdots y_{i_2} \cdot y_{i_1} = z_{i_r} \cdots z_{i_2} \cdot z_{i_1}$. (By Fact 5 the sequence of operators in β'_r which have superscript L [R] is

$i^L_{a_1} \cdot i^L_{a_o} \cdot I^L_{y_{i_r}}(a_o,a_1) \cdots I^L_{y_{i_2}}(a_o,a_1) \cdot I^L_{y_{i_1}}(a_o,a_1) \cdot E^L_{a_o} \cdot i^L_{a_o}$

$[i^R_{a_1} \cdot i^R_{a_o} \cdot I^R_{z_{i_r}}(a_o,a_1) \cdots I^R_{z_{i_2}}(a_o,a_1) \cdot I^R_{z_{i_1}}(a_o,a_1) \cdot E^R_{a_o} \cdot i^R_{a_o}]$. Note that $w_q = \alpha_{2n+4}(a_o,a_1)$ only cancels the two right-most and the two left-most operators in each of these sequences. Thus,

$I^L_{y_{i_r}}(a_o,a_1) \cdots I^L_{y_{i_2}}(a_o,a_1) \cdot I^L_{y_{i_1}}(a_o,a_1)$ and

$I^R_{z_{i_r}}(a_o,a_1) \cdots I^R_{z_{i_2}}(a_o,a_1) \cdot I^R_{z_{i_1}}(a_o,a_1)$ must be cancelled by operators in $w_{q-1} \cdots w_{r+2} \cdot w_{r+1}$. By Fact 5, $w_j \in \{\alpha_{2n+2+\sigma}(a_o,a_1): \sigma \in \{0,1\}\}$, for $r < j < q$. Note that, for $r < j < q$, if $w_j = \alpha_{2n+2+\sigma}(a_o,a_1)$ then the third operator from the left, in the sequence of operators in β'_j which have superscript L [R], is $i^L_{a_\sigma}$ $[i^R_{a_\sigma}]$. Furthermore, β'_{j+1} results from β'_j by omitting these two operators. Thus, the operator sequences $I^L_{y_{i_r}}(a_o,a_1) \cdots I^L_{y_{i_2}}(a_o,a_1) \cdot I^L_{y_{i_1}}(a_o,a_1)$ and $I^R_{z_{i_r}}(a_o,a_1) \cdots I^R_{z_{i_2}}(a_o,a_1) \cdot I^R_{z_{i_1}}(a_o,a_1)$ must be equal up to different superscripts implying that $y_{i_r} \cdots y_{i_2} \cdot y_{i_1} = z_{i_r} \cdots z_{i_2} \cdot z_{i_1}$.)

Thus, the sequence $i_1, i_2, \ldots, i_r$ is a solution to the PCP instance.

Q.E.D.

The result of Theorem 5 holds even for a subset of half-word protocols in which Σ is restricted to $\Sigma' \triangleq \{E^W_x, D^W_x, i^L_x, i^R_x, d^L_x, d^R_x: x \in N\}$, namely: