Abstract: A method (100) for encryption through a key driven polymorphic cipher by initializing a pseudorandom number generator, the method comprises the steps of generating a subkey table based on an encryption key (K) (102), extracting from the subkey table blocks of data X (104), mixing extracted data for a number of rounds (106) and producing a block of ciphertext (108). The method for generating a subkey table based on an encryption key (K) further comprises the steps of hashing the encryption key (K) to obtain h wherein h=hi(K) (202), placing h on first row of the subkey table (204), obtaining pi and q using h and from the encryption key (K) wherein pi and qi are two large secret integer numbers for i being 1 to n number of rounds (206), determining hi which is the subkey for i-th round (208) and placing hi as (i+1)st row of the subkey table (210).
FIELD OF INVENTION

The present invention relates to a method for encryption through a key driven polymorphic cipher.

BACKGROUND ART

Cryptography involves methods and principles that are used to transform data and hide its contents while establishing authenticity and preventing unauthorized access and/or modification of data. It uses mathematical algorithms to transform data into an unreadable format. Encryption using polymorphic cipher is discussed further in the present invention. Polymorphic cipher consists of a number of stacked primitive pseudo-random number generators that generate a confusion sequence, which is directly XOR-ed to plaintext bits.

Classical encryption mechanisms such as plaintext encryption are used as input to an encryption algorithm where the output is termed ciphertext. In some systems, however, multiple layers of encryption are used, in which case the ciphertext output of one encryption algorithm becomes the plaintext input to the next.

In addition, a cryptographic hash function, which is also known as a one-way encryption technique, is applied on several systems. In a one-way encryption technique, a string of text is only encoded, not decoded. The algorithms for one-way encryption are called hash algorithms. PHP uses the Message Digest (MD) hash algorithm, MD5, for one-way encryption. MD5 accepts a string as input and converts it to a unique 128-bit fingerprint of the message. MD5 is an irreversible process because it is not possible to decipher a message after it is converted into 128-bit fingerprint, which results a drawback of the encryption technique.

The test methodology of the present invention offers a paradigm of a key-driven polymorphic cipher wherein the resulting cipher is dynamically changed or morphed with each different user key through a series of stream cipher. The polymorphic cipher as disclosed in the present invention utilizes a variable word size and variable-size user
key. In the pre-processing stage, the user key is extended into a larger table or bit-level S-box using a specially developed hash-function. The generated table is used in a special configuration to substantially increase the substitution addressing space. Accordingly, this table is termed as the S-orb.

The proposed cipher of the present invention provides concepts of key-dependent number of rotations, key-dependent number of rounds and key-dependent addresses of substitution tables. Moreover, the parameters used to generate the different S-orb words are likewise key-dependent. The self-modifying proposed cipher, based on the aforementioned key-dependencies, provides an algorithm polymorphism and adequate security with a simple parallelizable structure.

The subject matter claimed herein is not limited to embodiments that solve any disadvantages or that operate only in environments such as those described above. Rather, this background is only provided to illustrate one exemplary technology area where some embodiments described herein may be practice.
SUMMARY OF INVENTION

In one embodiment of the present invention is a method for encryption through a key driven polymorphic cipher by initializing a pseudorandom number generator. The method comprises the steps of generating a subkey table based on an encryption key (K) (102), extracting from the subkey table blocks of data X (104), mixing extracted data for a number of rounds (106) and producing a block of ciphertext (108) characterized in that generating a subkey table based on an encryption key (K) further comprises the steps of hashing the encryption key (K) to obtain \( h_0 = h(K) \) (202), placing \( h_0 \) on first row of the subkey table (204), obtaining \( p_i \) and \( q_\iota \) using \( h_i = h(p_i \cdot h_{\iota M} + q_\iota) \) from the encryption key (K) wherein \( p_i \) and \( q_\iota \) are two large secret integer numbers for \( i \) being 1 to \( n \) number of rounds (206), determining \( h_i \) which is the subkey for \( i\)-th round (208) and placing \( h_i \) as \((i+1)st\) row of the subkey table (210).

The present invention consists of several novel features and a combination of parts hereinafter fully described and illustrated in the accompanying drawings, it being understood that various changes in the details may be made without departing from the scope of the invention or sacrificing any of the advantages of the present invention.
BRIEF DESCRIPTION OF ACCOMPANYING DRAWINGS

To further clarify various aspects of some embodiments of the present invention, a more particular description of the invention will be rendered by references to specific embodiments thereof, which are illustrated in the appended drawings. It is appreciated that these drawings depict only typical embodiments of the invention and are therefore not to be considered limiting of its scope. The invention will be described and explained with additional specificity and detail through the accompanying drawings in which:

FIG. 1 is a flowchart illustrating a method for encryption through a key driven polymorphic cipher.

FIG. 2 is a flowchart illustrating a method for generating a subkey table based on an encryption key (K).

FIG. 3 is a flowchart illustrating a method for extracting from the subkey table blocks of data X.

FIG. 4 is a flowchart illustrating a method for mixing extracted data for a number of rounds.
DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiments of the invention relate to a method for encryption through a key driven polymorphic cipher. Hereinafter, this specification will describe the present invention according to the preferred embodiments of the present invention. However, it is to be understood that limiting the description to the preferred embodiments of the invention is merely facilitating discussion of the present invention and it is envisioned without departing from the scope of the appended claims.

The present invention describes an encryption method though a key driven polymorphic cipher that utilizes a variable word size and variable-size user key. The polymorphic cipher utilizes a variable word size and variable-size user's key. In the preprocessing stage, the user key is extended into a larger table or bit-level S-box using a specially developed hash-function. The generated table, which is also known as the S-orb, is used in a special configuration to substantially increase the substitution addressing space. Accordingly, the proposed cipher provides concepts of key-dependent number of rotations, key-dependent number of rounds and key-dependent addresses of substitution tables. Moreover, the parameters used to generate the different S-orb words are likewise key-dependent.

Reference is being made to FIG. 1 and FIG. 2 collectively. FIG.1 is a flowchart illustrating a method for encryption through a key driven polymorphic cipher while FIG. 2 is a flowchart illustrating a method for generating a subkey table based on an encryption key (K). The method (100) for encryption through a key driven polymorphic cipher involves initialization of a pseudorandom number generator. The S-orb initialization is performed using the following recursive equation \( h_i = h(p, h_{i-1} + q) \) wherein \( h_1 \) is the hash function of the S-orb word \( (i) \). The total number of words of the S-orb \( (m) \) varies depending on the available memory and degree of security required. This value is taken equal to 6 resulting in an S-orb of six 192-bit words. The process is initialized with \( h_0 = h(k) \), where \( k \) is the user key, and \( p \) and \( q \) are two large secret integer numbers. These two numbers can also be obtained from the user key. The initial vector of the hash function \( (IV) \) is not necessarily to be kept secret.
Thererafter, a subkey table is generated based on an encryption key \((K)\) \((102)\). The process of generating a subkey table based on an encryption key \((K)\) further comprises the steps of hashing the encryption key \((K)\) to obtain \(h_0\) wherein \(h_0 = h(K)\) \((202)\), placing \(h_0\) on first row of the subkey table \((204)\), obtaining \(p_i\) and \(q_i\) using \(h_i = h(p_i \cdot h_{i-1} + q_i)\) from the encryption key \((K)\) wherein \(p_i\) and \(q_i\) are two large secret integer numbers for \(i\) being 1 to \(n\) number of rounds \((206)\), determining \(h_i\) which is the subkey for \(i\)-th round \((208)\) and placing \(h_i\) as \((i+1)\)st row of the subkey table \((210)\).

An assigned field in the round keys or S-orb words is further used to determine the location of the center, which is known as the "x-blocks" \((104)\). The operation is repeated by mixing extracted data for a number of rounds \((106)\) prior to producing a block of ciphertext \((108)\). The encryption key \((K)\) is used to determine \(n\) number of rounds and \(X\)-cell positions in the subkey table for each repetition while the subkey is used to determine selective addition and number of places for rotation in mixing extracted data for a number of rounds. Subsequently, the number of rounds is a number represented by a binary sequence from position 69 to position 71 of the encryption key.

Reference is now being made to FIG. 3. FIG. 3 is a flowchart illustrating a method for extracting from the subkey table blocks of data \(X\). The method for extracting from the subkey table blocks of data \(X\) further comprises selecting seven bits \(t\) from the encryption key beginning at the 23\textsuperscript{rd} position \((302)\), selecting seven bits \(t\) beginning from \((n+2)\)th place wherein \(n\) is the previous starting position \((304)\), locating an \(x\)-cell in the subkey table residing at memory address \(t\) \((306)\), producing an \(x\)-block which consist of \(x\)-cell and cells sharing an edge or vertex \((308)\) and allocating location weight 0 to 7 for cells in the \(x\)-block \((310)\).

The process of locating an \(x\)-cell in the subkey table residing at memory address \(t\) \((306)\) further comprises appending a last cell in a same column if there is no cell above the \(x\)-cell, appending a first cell of a same row if there is no cell to a right of the \(x\)-cell, appending the first cell of the same column if there is no cell below the located cell, appending the last cell of the same row if there is no cell to the left of the \(x\)-cell, appending the cell where a first column and a last row meets if there is no cell to a top right of the \(x\)-cell, appending the cell where a first row and the first column meets if there is no cell to a bottom right of the \(x\)-cell, appending the cell where the first row and a last
column meets if there is no cell at a bottom left of the x-cell and appending the cell where the last row and last column meets if there is no cell at the top left of the x-cell.

Reference is now being made to FIG. 4. FIG. 4 is a flowchart illustrating a method for mixing extracted data for a number of rounds. The selective exclusive-or operation is performed to realize the required homophonic substitution. The mixing of extracted data for a number of rounds further comprises the steps of performing exclusive-or on \((i+j)\)th bit of extracted data and \((i+j)\)th bit of a subkey for every octet in the extracted data which begin at position \(i\) in the data resulting in \(C1\) which is the value at position \((i + j)\) if bit at location weight \(j\) in x-block has value 1 otherwise the \((i+j)\)th bit of \(C1\) takes the value of the \((i+j)\)th bit of the data (402).

Thereafter, \(C1\) is partitioned into at least 6, 12 or 24 words of the same length \(L\) (404) while every word is rotated to a left by \(m\) places wherein \(m\) is the number represented by a binary sequence of length \(\log_2L\) taken from position 16 of the subkey resulting in \(C2\) (406). Subsequently, the exclusive-or operation is performed on \((i+j)\)th bit of \(C2\) and \((i+j)\)th bit of the subkey for every octet in \(C2\) which begin at position \(i\) in \(C2\) resulting in \(C3\) which is the value at location \(i+j\) if bit at location weight \(j\) in x-block has value 0 otherwise the \((i+j)\)th bit of \(C3\) takes the value of the \((i+j)\)th bit of \(C2\) (408) and performing exclusive-or on \(C3\) and its subkey (410). Finally, a block of ciphertext is produced upon mixing the extracted data for the said number of rounds. The process of producing a block of ciphertext further comprises computing a probability of a successful attack on the block of ciphertext from shuffled \(n\) blocks. The block of ciphertext is applicable to software and hardware based application such as voice and image encryption.

The methodology for encryption through a key driven polymorphic cipher is an absolute enhancement towards data security as it prevents attacks with the features on the hash function. The performance on modern superscalar processors of the hash function of the present invention were optimized and verified.

The present invention may be embodied in other specific forms without departing from its essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore indicated by the appended claims rather than by the foregoing description. All changes,
which come within the meaning and range of equivalency of the claims, are to be embraced within their scope.
CLAIMS

1. A method (100) for encryption through a key driven polymorphic cipher by initializing a pseudorandom number generator, the method comprises the steps of:
   - generating a subkey table based on an encryption key (K) (102);
   - extracting from the subkey table blocks of data X (104)
   - mixing extracted data for a number of rounds (106);
   - and producing a block of ciphertext (108)

characterized in that

   generating a subkey table based on an encryption key K further comprises the steps of:
   - hashing the encryption key (K) to obtain \( h_0 \) wherein \( h_0 = h(K) \) (202);
   - placing \( h_0 \) on first row of the subkey table (204);
   - obtaining \( p_i \) and \( q_i \) using \( h_i = h(p_i \cdot h_m + q_i) \) from the encryption key (K) wherein \( p_i \) and \( q_i \) are two large secret integer numbers for \( i \) being 1 to \( n \) number of rounds (206);
   - determining \( h_i \) which is the subkey for \( i \)-th round (208); and
   - placing \( h_i \) as \( (i+1)\)st row of the subkey table (210)

2. The method (100) according to claim 1, wherein the encryption key (K) is used to determine \( n \) number of rounds and X-cell positions in the subkey table for each repetition.

3. The method (100) according to claim 1, wherein the subkey is used to determine selective addition and number of places for rotation in mixing extracted data for a number of rounds.

4. The method (100) according to claim 1, wherein the number of rounds is a number represented by a binary sequence from position 69 to position 71 of the encryption key (K).
5. The method (100) according to claim 1, wherein extracting from the subkey table blocks of data $X$ further comprises:

- selecting seven bits $t$ from the encryption key beginning at the $23^{rd}$ position (302);
- selecting seven bits $t$ beginning from $(n+2)^{th}$ place wherein $n$ is the previous starting position (304);
- locating an $x$-cell in the subkey table residing at memory address $t$ (306);
- producing an $x$-block which consist of $x$-cell and cells sharing an edge or vertex (308); and
- allocating location weight 0 to 7 for cells in the $x$-block (310).

6. The method (100) according to claim 51, wherein locating a cell in the subkey table residing at memory address $t$ further comprises:

- appending a last cell in a same column if there is no cell above the $x$-cell;
- appending a first cell of a same row if there is no cell to a right of the $x$-cell;
- appending the first cell of the same column if there is no cell below the located cell;
- appending the last cell of the same row if there is no cell to the left of the $x$-cell;
- appending the cell where a first column and a last row meets if there is no cell to a top right of the $x$-cell;
- appending the cell where a first row and the first column meets if there is no cell to a bottom right of the $x$-cell;
- appending the cell where the first row and a last column meets if there is no cell at a bottom left of the $x$-cell; and
- appending the cell where the last row and last column meets if there is no cell at the top left of the $x$-cell.
7. The method (100) according to claim 1 wherein mixing extracted data for a number of rounds further comprises:

performing exclusive-or on (i+j)th bit of extracted data and (i+j)th bit of a subkey for every octet in the extracted data which begin at position i in the data resulting in C_1 which is the value at position (i + j) if bit at location weight j in x-block has value 1 otherwise the (i+j)th bit of C_1 takes the value of the (i+j)th bit of the data (402);

partitioning C_1 into at least 6, 12 or 24 words of the same length L (404);

rotating every word to a left by m places wherein m is the number represented by a binary sequence of length \log_2 L taken from position 16 of the subkey resulting in C_2 (406);

performing exclusive-or on (i+j)th bit of C_2 and (i+j)th bit of the subkey for every octet in C_2 which begin at position i in C_2 resulting in C_3 which is the value at location (i+j) if bit at location weight j in x-block has value 0 otherwise the (i+j)th bit of C_3 takes the value of the (i+j)th bit of C_2 (408); and

performing exclusive-or on C_3 and its subkey (410).

8. The method as claimed in claim 1 wherein the block of ciphertext is applicable to software and hardware based application.

9. The method as claimed in claim 1, wherein producing a block of ciphertext further comprises computing a probability of a successful attack on the block of ciphertext from shuffled n blocks.
Encryption key, K → Subkey table generation → Subkey for the round

Subkey table

7 bits of K → x-block extraction → x-block for the round

A block of data, X → Shuffling/mixing

CI

Reached number of rounds?

Yes → A block of ciphertext, Y := CI

No → CI
200

202
Read user key K to obtain $h_0$ wherein $h_0 = h(K)$

204
Write in sub-key file

206
From user key K
Get p, q

208
$h_{i+1} = h(h_i^* p + q)$

210
$i = i + 1$

$\diamondsuit i \geq i_{\text{max}} ?$

\text{Halt}

FIG. 2
300

302 Select seven bits t from encryption key

304 Select seven bits t beginning from (n+2)th place

306 Locate an x-cell in the subkey table

308 Produce an x-block

310 Allocate location weight 0 to 7 for cells in x-block

FIG. 3
Perform exclusive-or (i+j)th bit of extracted data resulting in C1

Partition C1 into at least 6, 12 or 24 words

Rotate every word to a left by m places resulting in C2

Perform exclusive-or on (i+j)th bit of C2 resulting in C3

Perform exclusive-or on C3

FIG. 4
A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl
H04L 9/26 (2006 01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPODOC, WPI, TXTUS1, TXTUS2, TXTUS3, TXTUS4, TXTTEPI, TXTGBI, TXTWOI with IPC marks and keywords including polymorphic cipher, subkey, key schedule, key expansion, key driven/dependent.

Google Scholar search with similar keywords as above.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
</table>

X Further documents are listed in the continuation of Box C  x See patent family annex

Date of the actual completion of the international search 06 September 2010
Date of mailing of the international search report 09 SEP 2010

Name and mailing address of the ISA/AU
AUSTRALIAN PATENT OFFICE
PO BOX 200, WODEN ACT 2606, AUSTRALIA
E-mail address: pct@ipaustralia.gov.au
Facsimile No. 61 2 6283 7999

Authorized officer
MLADEN MITIC
AUSTRALIAN PATENT OFFICE
(ISO 9001 Quality Certified Service)
Telephone No: 61 2 6283 3193

Form PCT/ISA/2 10 (second sheet) (July 2009)
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 6259789 B1 (PAONE) 10 January 2001 - Whole document</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>US 725423 1 B1 (VAN DYKE et al.) 7 August 2007 - Whole document</td>
<td></td>
</tr>
</tbody>
</table>
This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>EP 1172965</td>
<td>CN 1349182 JP 2002023622 KR 20020006475</td>
</tr>
<tr>
<td></td>
<td>US 2002021802 US 7194090</td>
</tr>
<tr>
<td>US 6259789</td>
<td>NONE</td>
</tr>
<tr>
<td>US 7254231</td>
<td>NONE</td>
</tr>
<tr>
<td>US 2006126835</td>
<td>KR 20060066348</td>
</tr>
</tbody>
</table>

Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001.

END OF ANNEX