VHDL Processor for Covering Information Using MD5 Algorithm I

معالج (VHDL) لتغطية المعلومات بأستخدام خوارزمية MD5 I

Ass. Prof. Thamir R. Saeed **Eng. Hashmea S. Dakel**
University of Technology, Baghdad

Eng. Najmah A. beeb
Al-Mansour Univ. College, Baghdad

Lecture Ivan A. Hashim **Prof. Jawad K. Ali**
University of Technology, Baghdad

VHDL Processor for Covering Information Using MD5 Algorithm I

معالج (VHDL) لتغطية المعلومات بأستخدام خوارزمية MD5 I

Ass. Prof. Thamir R. Saeed **Eng. Hashmea S. Dakel**
University of Technology, Baghdad

Eng. Najmah A. beeb
Al-Mansour Univ. College, Baghdad

Lecture Ivan A. Hashim **Prof. Jawad K. Ali**
University of Technology, Baghdad

Abstract

The data protection means a life protection, since the life has become some kind of data transfer. In the data transformation, there are two factors that must be satisfied; distortion and security at any joint or destination point. The work of this paper will focus on the security factor. The efficiency of the MD5 algorithm urged us to use it to cover the information which needs to be secure. MD5 represents one way security algorithm with covering algorithms that are popular ways for hidden the data. In this paper we use MD5 with many covering algorithms in four scenarios which proposed to increase the degree of security. The scenarios operation shows efficient results for our goal. Therefore, two power points can gain from this combination (MD5 and proposed cover algorithms), which are the length of the stream before it has been repeating and its complexity. The stream length for the proposed algorithms with respect to the MD5 will be increased form 256 for 8-bit in MD5 to 16384 for scenarios B and C and 49152 for scenario D with 20 nsec and 60 n sec respectively. The covering and

recovering circuits have been implemented using Xilinx Spartan 3-xc3s1400a-4fg484.

Key words: VHDL, FPGA-Spartan-3, Code-Cover, and MD5.

1. Introduction

In the data transformation, there are two factors must be satisfied one hidden existence of the information (message or data) and the other distorts the massage itself, because, it is desired to communicate data with high-security [1]. Also, the security must have an associated factor is the authentication. According to this, various types of algorithms have to provide high security for information on controlled networks. Therefore, there are many properties in security mechanism for broadcast traffic in a single-hop wireless sensor network have to be taken into account; these are [2,3];

- *Confidentiality (or Privacy).*
- *Authenticity.*
- *Freshness.*
- *Semantic Security.*
- *Dynamic Data.*
- *Delay Tolerance.*
- *Incremental Processing.*
- *Resilience to Loss.*
- *Integrity.*
- *Identification.*
- *Nonrepudiation.*
- *Access control.*

Protecting access to information for reasons of security is still a major reason for using secured algorithms [1]. While, not all

algorithms provide all these properties, but the algorithm which wants to achieve the purpose of it must achieve most of them.

In the existing technical literature, many related studies on data security have been reported. It is clear from this review that only one work has used the MD5 for error correction but not for covering.

In [2], time-varying keys (based on a key-chain) for broadcast encryption are used, and extend the basic key-chain mechanism to incorporate limited protection against key loss. While [4], described a new class of lightweight, symmetric-key digital certificates called extended TESLA certificates and a source authentication protocol for wireless group communication that is based on the certificate. A protocol which provides a three cryptographic primitives, such as integrity, confidentiality and authentication with the help of Elliptic Curve Cryptography, Dual-RSA algorithm and Message Digest MD5 has been presented in [1]. The work presented in [5] has calculated the message digest from MD5 algorithm and propagated its output through the communication medium along with the original message from the sender side, and on the receiver side integrity of the message can be verified by recalculating the message digest of the received message and comparing the two digest values. To overcome the handover overhead and, hence minimize authentication time, a new secure MIH message transport solution, referred as Media Independent Handover (MIHSec) has been reported in [6]. In [7], a family of fast and provably secure cryptographic hash functions was proposed. The security of these functions relies directly on the well-known syndrome decoding problem for linear codes.

Therefore, there are many algorithms in that field; each one operates at the transfer data from a viewpoint which related to the overall system. In the present work, we will use the algorithm which already uses for password, these are MD5, and this usage for data or information covering is our work

contribution. The reason for using this algorithm is because of that fact that it is one way algorithm, i.e., one cannot predict any code of MD5 from the previous one.

2. Message digests (MD)

Message digest (MD) algorithms, also called as Hash algorithms, generate a unique message digest for an arbitrary message. Furthermore, it's used widely in cryptographic protocols and Internet communication [8]. One of the most famous variants is the MD5 message digest algorithm developed by Ronald Rivest [9]. The message digest to be generated by MD5 algorithm has the irreversible and non-counterfeit features, so MD5 algorithm is superior in anti-tamper capability. In addition, it can be considered as a fingerprint of the message, and it must have the following properties:

First - must be easy to compute,

Second - it must be very hard to compute the message from the digest and,

Third - it must be hard to find another message which has the same message digest as the first one [9].

The core of MD algorithm is a hash function, which compresses a string of arbitrary length into a string of fixed length. It provides a unique relationship between the input and the hash value and hence replaces the authenticity of a large amount of information (message) by the authenticity of a much smaller hash value (authenticator) [10].

3. The proposed Algorithm

As stated in the introduction, the work in the data transformation field has two important factors, distortion and security. The work of this paper will focus on the security factor, and upto author’s knowledge it is novel.

Our approaches to data secrecy in this paper are the use of the MD5 core final and iterations output as a set of successive

keys derived from the repeated one-way hashing of an initial key code for covering the required information (data or massage). In our contribution, one can see how a key can be used for covering a certain message to ensure secrecy, authenticity, replay protection (freshness), and high message entropy (i.e., cipher messages does not repeat even if the plain-text messages do). We also highlight several natural advantages of our approach, such as the ability to accommodate dynamic data, as well as protection against compromised keys. The implementation has been carried out using Xilinx Spartan 3-xc3s1400a-4fg484.

The general formula for the message which can be used is [5];

$$Message=((a\text{-}z)+(A\text{-}Z)+(0\text{-}9)+(!\text{-};))^{m} \quad (1)$$

The message digest should be a string of fixed length.

$$MessageDigest=((0\text{-}9)+(A\text{-}F))^{n} \quad (2)$$

Where

n- a fixed natural number MD5 iteration (1-64) and related to the input code.

M- number of bits which will represent the code.

We can use the MD5 properties to cover the wanted data in the communication system. Our work is based on the following requirements:

1. Store numbers of codes up to 256 (more can be stored) 8-bit data in the RAM memory (Key) code (DC) which will be entering to the MD5 core.
2. MD5 core output is 128-bit after 64 iterations, for each iteration also MD5 has been 128-bit as the output.
3. The output of the final iteration of MD5 or any selected iteration can make XOR or other process with information and then transmits it in the certain stream as in scenarios.

Four scenarios can represent our work in a transmitter side. These are;

1.1 Scenario A (SA)

1. Select the DC from Stored (RAM).
2. Enter DC to the MD5 core.
3. Then, made XOR between the output of the MD5 (final output after 64 iteration (MO-128 bit)) and the information (I (128 bit)) the resultant (MOI).

$$MOI=MO \ XOR \ I \qquad (3)$$

4. The stream output of this scenario is shown in figure(1)
5. The receiver side separates the first 8-bit which represent the DC address , then the code will be entered to the MD5 in the receiver.
6. Separate the 128-bit after 8-bit and made XOR with an output of the MD5 output. The result will be the information.

$$I=MOI \ XOR \ MO \qquad (4)$$

1.2 Scenario B (SB)

1. Select the DC from Stored (RAM).
2. Enter DC to the MD5 core.
3. Select certain iteration (CI) from the MD5 processing to deal with its output (MD5I) MO(i).
4. Then, made XOR between (MD5I) and the information (I) , the resultant is (MOI(i)).

$$MOI(i)=MO \ (i) \ XOR \ I \qquad (5)$$

5. The stream output will be as in figure(2)

1.3 Scenario C (SC)

The selection of code address of the scenario A or code address and iteration number of the scenarios' B is made randomly.

$$DC = Rand(k) \quad (6)$$

k- address (size) of RAM which DC was stored.

Then same process will follow as that of scenario B

On the receiver side, the stream will be received and then separated according to the transmitted scenario. When scenario A was applied afterwards separates the first 8-bit which represent the MD5 code address. Next, this code has been used to generate the MD5 output (128-bit), after that made XOR between the MD5 output and the rest 128-bits of the received stream to obtain the wanted information. When scenario B was applied subsequently separates the first 8-bits which represent the MD5 code address and the other 8-bits which represent the MD5 iteration. Later, enter the code to the MD5 core and compare the iteration number with received one to obtain the output of MD5 at certain iteration then made XOR between this output and the rest 128-bit from the received stream to obtain the wanted information.

1.4 Scenario D (SD)

This scenario is a combination of the two scenarios B and C and adds to them, the transmitted stream is repeated three times with different code and iteration number, (can make more than three times if wanted). This repetition is used for error detection where, compares the output of each one if it is equal, then it is correct if it is not, this mean an error occurred.

Figure(3) represents a covering / recovering proposed algorithm flowchart, which represents the scenario C. It is clear the random block selects the Key-code to the MD5 core and certain iteration for making XOR with information. Then put the address of Key-code, number of iterations and XOR output in

the stream to transmit, while, on the receiver side as explained above. Figure (4) represents the VHDL core program which downloaded in Xilinx Spartan 3-Xc3s1400-4fg484, to accomplish the process of the proposed algorithm. Figure(5) represents the covering and recovering code streams, where, at the covering (transmitter) side, the code stream as the output of MD5 for certain Key code, has the address $(05)_{Hex}$ and the iteration number $(13)_{Hex}$ and made XOR with information, which is represented as MSG[127:0] ([128-bit in Hex(00000005000000000000000000000000) 32 digits each one 4-bit]). The covered Data as in Figure (5-a) which represents a stream [143:0](144-bit in Hex (0513EBFC3BA7F92388B666DAE96F430F0A2D) 36 digits each one 4-bit). The first two digits Hex (05) represents the Key-code address to the MD5. The second two digits Hex (13) represents the iteration number. The remaining 32-digits(128-bits) represent the output of the XOR process between the information which we want to cover and the output of the MD5 at Hex(13) iteration when using key-code Hex (05). At the receiver side, receive the stream (covering data) as in Figure(5-b), and separate first two digits and consider it as the address of the Key-code to the MD5. After that, the second two digits as the iteration number which compared it with the iteration counter to reach for it. Then, made XOR between the output of MD5 at certain iteration $(13)_{Hex}$ from the Key-code, which has been addressed $(05)_{Hex}$ with rest 32 digits to extract the information as it's clear in the Figure.

Figure (5) represents scenario D, where the message has been covered three times with different Key-code and iteration number as shown in Figure (6-a and b). At the receiver side, receive the three messages, then, separate the Key-code address and iteration number to extract the information and then, compare them if it is equal this meaning no error, as in Figure (6-c and d).

We repeat these operations for many cases and with random selection of Key-code and number of iterations, and we get the

same result and without error. The bit rate or information bit in the stream and the operation time will be depending on the scenario selection, and the number of repetition transmits operation as in Tables (1).

4. Conclusions:

The MD5 represents one of the efficient algorithms for making a password. For this reason it has been used to covering the information gives a power to the first point of properties (Privacy). The use of MD5 increases the number of combinations from 256 to 16384 and with repetition of the 49152 for each 128-bits for covering the information and can increase this number of combinations. The time of covering as in Table(1) is 20 nsec but the time of the attack will take many years and anyone can calculate it. Using of The Xilinx technology gives a power to this subject by reducing the process time and give a flexibility of changing or improving the design

As a future work, In the next paper, we will use MD5 for hashing the information and then rehashing it by adopting new algorithms.

Figure(1) Transmit a stream of SA

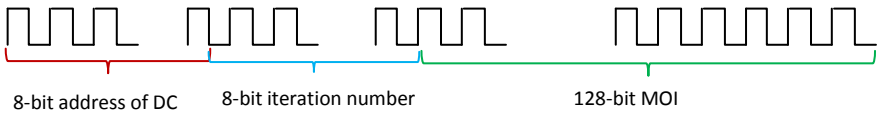


Figure (2) Transmit a stream of SB

Table (1) The Proposed Scenarios specification

Scenario	Combination	% with respect to MD5 (MD5 combination = 265)	No. of information bits in Tx stream	Process time for covering information
A	256	Equal	128	
B	16384	64 % over	128	20 nsec
C	16384	64 % over	128	20 nsec
D	49152	192 % over	128	60 nsec

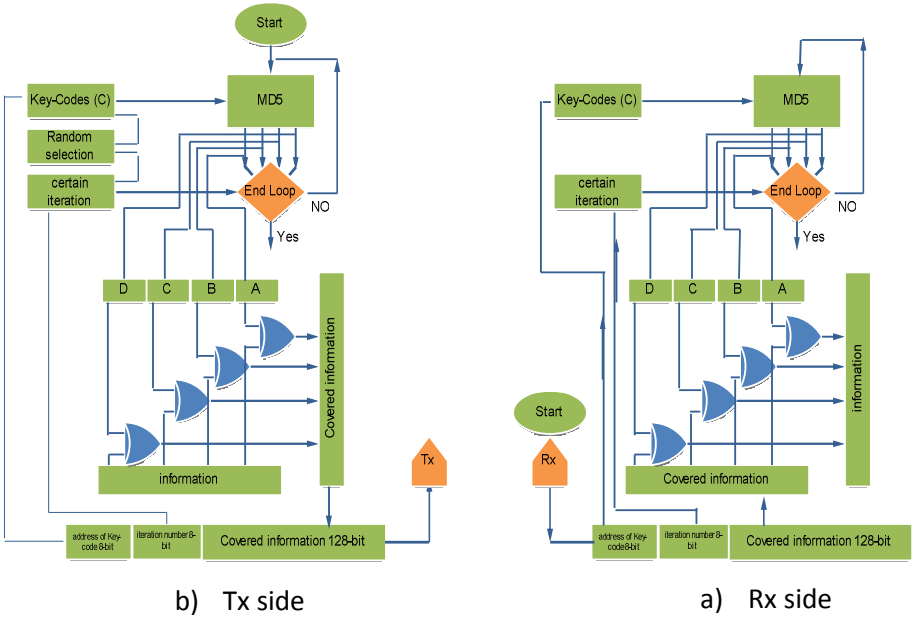


Figure (3) Proposed algorithm flowchart

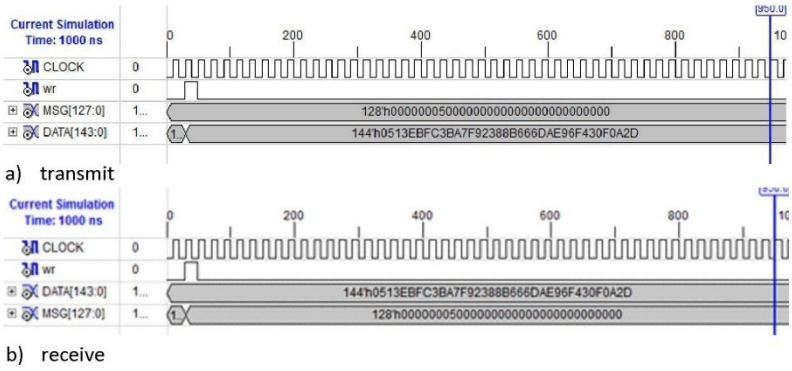
```
MD5 Core
    AA:=A;BB:=B;CC:=C;DD:=D;
    for i in 0 to 63 loop
      if (i>=0) and (i<=15)then
         F:=((B and C)or((not B) and D));
      elsif (i>=16) and (i<=31)then
         F:=((B and D)or(B and (not D)));
      elsif (i>=32) and (i<=47)then
         F:=(B xor C xor D);
      elsif (i>=48) and (i<=63)then
         F:=(C xor (B and (not D)));
      end if;
      TS(31-S(i)downto 0):=T(i)(31 downto S(i));
      TS(31 downto (31-S(i)+1)):=T(i)((S(i)-1) downto 0);
      A:=B+((A+F)+X(K(i))+TS);
      D:=C;C:=B;B:=A;A:=D;
      if i=IT then
        At:=A+AA;Bt:=B+BB;Ct:=C+CC;Dt:=D+DD;
        Git(31 downto 0):=At;
        Git(63 downto 32):=Dt;
        Git(95 downto 64):=Ct;
        Git(127 downto 96):=Dt;
      end if;
    end loop;
    A:=A+AA;B:=B+BB;C:=C+CC;D:=D+DD;
    G(31 downto 0):=A;
    G(63 downto 32):=B;
    G(95 downto 64):=C;
    G(127 downto 96):=D;
XOR Process to covering the information
    GREG:=DREG xor Git;
Tx Stream arrange
    DATA(127 downto 0)<=GREG;
    DATA(135 downto 128)<=CONV_STD_LOGIC_VECTOR(IT,8);
    DATA(143 downto 136)<=CONV_STD_LOGIC_VECTOR(BLIind,8);
    end Behavioral;
```

a) Tx Side VHDL Prog.

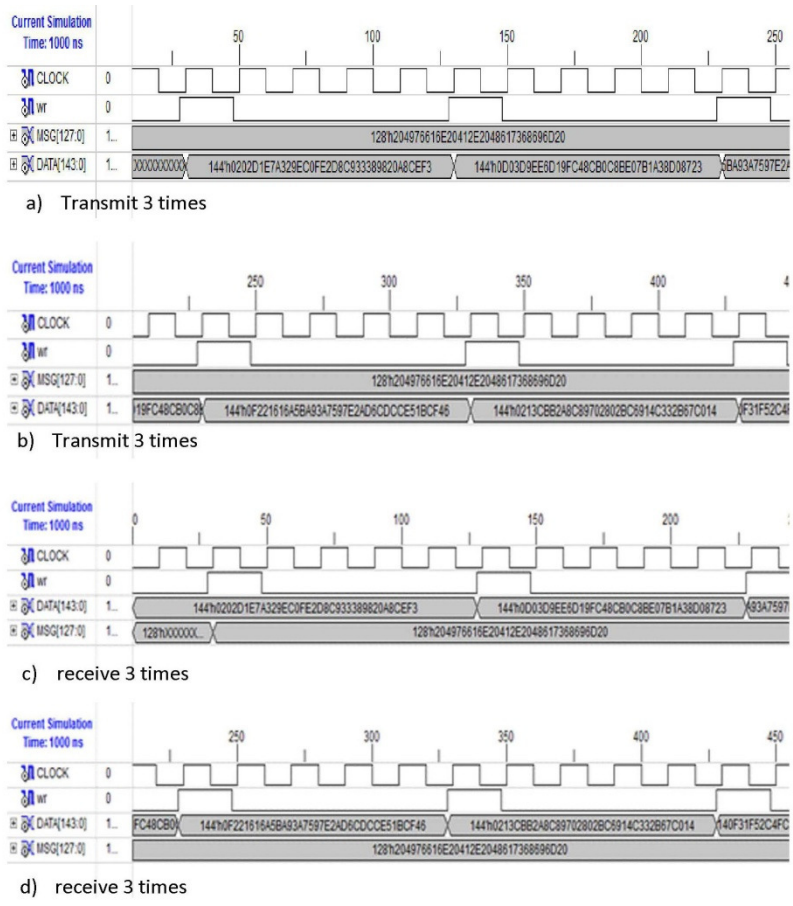
```
Rx Stream Arrange
   DATA:in std_logic_vector(143 downto 0);
   MSG:out std_logic_vector(127 downto 0)
   variable G,Git:std_logic_vector(127 downto 0);
   variable X:y3;
   BLIind:=CONV_INTEGER(DATA(143 downto 136));
   IT:=CONV_INTEGER(DATA(135 downto 128));
   DREG:=DATA(127 downto 0);
   BL(L-1 downto 0):=CONV_STD_LOGIC_VECTOR(BLI(BLIind),32);
   BL(L):='1';
   BL(n downto (n-63)):=CONV_STD_LOGIC_VECTOR(L,64);
   for i in 0 to 15 loop
     X(i):= BL((i*32)+31 downto (i*32));
   end loop;
Same MD5 Core
XOR Process Extract the information
135 GREG:=DREG xor Git;
136 MSG(127 downto 0)<=GREG;
```

b) Rx Side VHDL Prog.

Figure (4) Tx/Rx VHDL core Prog. For proposed algorithm



Figure(5) Tx/Rx code stream



Figure(6) two samples of three transmit/ receive scenario

References:

[1] S. Subasree and N. K. Sakthivel, " Design a New Security Protocol Using Hybrid Cryptography Algorithm", IJRRAS , Vol. 2 , No. 2, 2010.

[2] V. Sivaraman, D. Ostry, J. Shaheen, A. J. Hianto, and S. Jha, " Broadcast Secrecy via Key-Chain-Based Encryption in Single-Hop Wireless Sensor Networks", EURASIP Journal on Wireless Communications and Networking, 2011.

[3] M. Wolf, A. Weimerskirch, and T. Wollinger, " State of the Art: Embedding Security in Vehicles", EURASIP Journal on Embedded Systems, 2007.

[4] A. Roy-Chowdhury and J. S. Baras, " Energy-Efficient Source Authentication for Secure Group Communication with Low-Powered Smart Devices in Hybrid Wireless/Satellite Networks", EURASIP Journal on Wireless Communications and Networking , 2011.

[5] Prof. R. Mohanty, N. Sarangi, and S. K. Bishi, " A Security Cryptographic Hashing Algorithm", arXiv, 2010.

[6] J. Won, M. Vadapalli, C. Cho, and V. C. M. Leung, " Secure Media Independent Handover Message Transport in Heterogeneous Networks", EURASIP Journal onWireless Communications and Networking, 2009.

[7] D. Augot, M. Finiasz, and N. Sendrier, " A Fast Provably Secure Cryptographic Hash Function", Citeseer,IVSL, 2009**.**

[8] Dongjing H. and Zhi X., " Multi-parallel Architecture for MD5 Implementations on FPGA with Gigabit-level Throughput", 2010 International Symposium on Intelligence Information Processing and Trusted Computing.

[9] Kimmo J., Matti T. and Jorma S., " Hardware Implementation Analysis of the MD5 Hash Algorithm", Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.

[10] Janaka D., Howard M. H. and Venkatesan R., " FPGA Implementation of MD5 Hash Algorithm", IEEE, Electrical and Computer Engineering Canadian Conference,IVSL 2001.

الخلاصة:ـ

معالج (VHDL) **لتغطية المعلومات بأستخدام خوارزمية** MD5 I

حمايةِ البياناتَ تعني حمايةِ الحياةِ, لان الحياة اصبحت جانبا من المعلومات المتنقله. في عملية نقل البياناتَ، هناك عاملان يجب أنْ يٌراعا هما؛ التشويه والأمن بأي نقطة ربط أو مفصلَ. عمل هذا البحث متُركّزُ على عاملِ الأمنَ. كفاءة خوارزميةَ (إم دي ٥) حَثّتْنا لإسْتِعْمالها لتَغْطية المعلوماتِ المراد لها ان تكُونَ آمنةَ. في هذا البحث استخدمنا إم دي ٥ مع عدة خوارزمياتِ تغطية المعلومات ضمن اربع سيناريوهات لزيادة درجة الامن. أظهرت هذه السيناريوهات نتائج كفوءة. نقطتي قوة من دمج (إم دي ٥) مع خوارزمياتَ التغطية هما، طول سلسلة الترميزِ قبل تكرّرها والتعقيدِ . طول السلسلة لخوارزمياتِنا المُقتَرَحةِ نسبة الى إم دي ٥ والتي طولها ٢٥٦ مره لقطعةِ ٨ بت بينما لسيناريوهاتنا المقترحة هو ١٦٣٨٤ للسيناريو بي وسي و٤٩١٥٢ للسيناريو دي ب٢٠ nsec و ٦٠ n sec على التعاقب. إنّ بناء دوائر الغطاء ودوائرِ فتح الغطاء بٌنيا بإستعمال 3-xc3s1400 a-4fg 484 Xilinx Spartan