Novel Algorithm using Hybrid Architecture for High Security.

N Krishna Chandu and Dr. M. Gurunadha Babu

Abstract: A Computer Network is an interconnected group of autonomous computing nodes, which use a well defined, mutually agreed set of rules and conventions known as protocols, interact with one-another meaningfully and allow resource sharing preferably in a predictable and controllable manner. Communication has a major impact on today's business. It is desired to communicate data with high security. Security Attacks compromises the security and hence various Symmetric and Asymmetric cryptographic algorithms have been proposed to achieve the security services Such as Authentication, Confidentiality, Integrity, Non-Repudiation and Availability.

At present, various types of cryptographic algorithms provide high security to information on controlled networks. These algorithms are required to provide data security and users authenticity. To improve the strength of these security algorithms, a new security protocol for on line transaction can be designed using combination of both symmetric and asymmetric cryptographic techniques. This protocol provides three cryptographic primitives such as integrity, confidentiality and authentication. These three primitives can be achieved with the help of Elliptic Curve Cryptography, Dual-RSA algorithm and Message Digest MD5.

That is it uses Elliptic Curve Cryptography for encryption, Dual-RSA algorithm for authentication and MD-5 for integrity. This new security protocol has been designed for better security with integrity using a combination of both symmetric and asymmetric cryptographic techniques.

Keywords: ECC, Dual-RSA, MD5, DES.

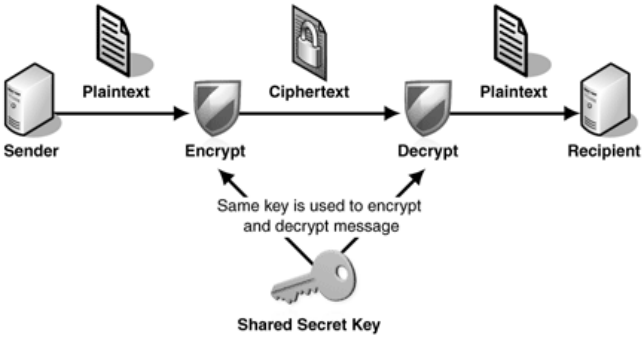
I. INTRODUCTION

2.1 Modern Cryptography

The modern field of cryptography can be divided into several areas of study. The chief ones are discussed here.

2.2 Symmetric-key cryptography

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). This was the only kind of encryption publicly known until June 1976.



The modern study of symmetric-key ciphers relates mainly to the study of block ciphers and stream ciphers and to their applications. A block cipher is, in a sense, a modern embodiment of Alberta's poly alphabetic cipher: block ciphers take as input a block of plaintext and a key, and output a block of cipher text of the same size. Since messages are almost always longer than a single block, some method of knitting together successive blocks is required. Several have been developed, some with better security in one aspect or another than others. They are the modes of operation and must be carefully considered when using a block cipher in a cryptosystem.

The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are block cipher designs which have been designated cryptography standards by the US government (though DES's designation was finally withdrawn after the AES was adopted). Despite its deprecation as an official standard, DES (especially its still-approved and much more secure triple-DES variant) remains quite popular; it is used across a wide range of applications, from ATM encryption to e-mail privacy and secure remote access. Many other block ciphers have been designed and released, with considerable variation in quality. Many have been thoroughly broken. See Category: Block ciphers.

Stream ciphers, in contrast to the 'block' type, create an arbitrarily long stream of key material, which is combined with the plaintext bit-by-bit or character-by-character, somewhat like the one-time pad. In a stream cipher, the output stream is created based on a hidden internal state which changes as the cipher operates. That internal state is initially set up using the secret key material. RC4 is a widely used stream cipher; see Category: Stream ciphers. Block ciphers can be used as stream ciphers; see Block cipher modes of operation.

Cryptographic hash functions are a third type of

cryptographic algorithm. They take a message of any length as input, and output a short, fixed length hash which can be used in (for example) a digital signature. For good hash functions, an attacker cannot find two messages that produce the same hash. MD4 is a long-used hash function which is now broken; MD5, a strengthened variant of MD4, is also widely used but broken in practice. The U.S. National Security Agency developed the Secure Hash Algorithm series of MD5-like hash functions: SHA-0 was a flawed algorithm that the agency withdrew; SHA-1 is widely deployed and more secure than MD5, but cryptanalysts have identified attacks against it; the SHA-2 family improves on SHA-1, but it isn't yet widely deployed, and the U.S. standards authority thought it "prudent" from a security perspective to develop a new standard to "significantly improve the robustness of NIST's overall hash algorithm toolkit." Thus, a hash function design competition is underway and meant to select a new U.S. national standard, to be called SHA-3, by 2012.

Message authentication codes (MACs) are much like cryptographic hash functions, except that a except that a secret key is used to authenticate the hash value on receipt.

2.3 Public-key cryptography

Symmetric-key cryptosystems use the same key for encryption and decryption of a message, though a message or group of messages may have a different key than others. A significant disadvantage of symmetric ciphers is the key management necessary to use them securely. Each distinct pair of communicating parties must, ideally, share a different key, and perhaps each cipher text exchanged as well. The number of keys required increases as the square of the number of network members, which very quickly requires complex key management schemes to keep them all straight and secret. The difficulty of securely establishing a secret key between two communicating parties, when a secure channel doesn't already exist between them, also presents a chicken-and-egg problem which is a considerable practical obstacle for cryptography users in the real world.

Whitfield Diffie and Martin Hellman, authors of the first paper on public-key cryptography

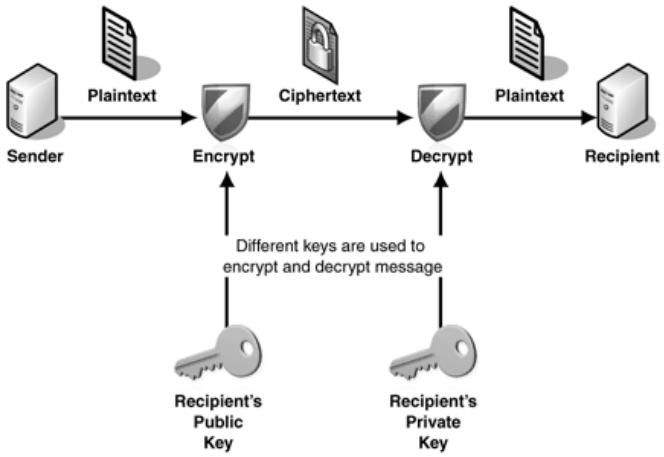
In a groundbreaking 1976 paper, Whitfield Diffie and Martin Hellman proposed the notion of public-key (also, more generally, called asymmetric key) cryptography in which two different but mathematically related keys are used — a public key and a private key. A public key system is so constructed that calculation of one key (the 'private key') is computationally infeasible from the other (the 'public key'), even though they are necessarily related. Instead, both keys are generated secretly, as an interrelated pair. The historian David Kahn described public-key cryptography as "the most revolutionary new concept in the field since poly alphabetic substitution emerged in the Renaissance".

In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. The public key is typically used for encryption, while the private or secret key is used for decryption. Diffie and Hellman showed that public-key cryptography was possible by presenting the Diffie-Hellman key exchange protocol.

In 1978, Ronald Rivest, Adi Shamir, and Len Adleman invented RSA, another public-key system.

In 1997, it finally became publicly known that asymmetric key cryptography had been invented by James H. Ellis at GCHQ, a British intelligence organization, and that, in the early 1970s, both the Diffie-Hellman and RSA algorithms had been previously developed (by Malcolm J. Williamson and Clifford Cocks, respectively).

The Diffie-Hellman and RSA algorithms, in addition to being the first publicly known examples of high quality public-key algorithms, have been among the most widely used. Others include the Cramer-Shoup cryptosystem, ElGamal encryption, and various elliptic curve techniques.
Padlock icon from the Firefox Web browser, meant to indicate a page has been sent in SSL or TLS-encrypted protected form. However, such an icon is not a guarantee of security; any subverted browser might mislead a user by displaying such an icon when a transmission is not actually being protected by SSL or TLS.



In addition to encryption, public-key cryptography can be used to implement digital signature schemes. A digital signature is reminiscent of an ordinary signature; they both have the characteristic that they are easy for a user to produce, but difficult for anyone else to forge. Digital signatures can also be permanently tied to the content of the message being signed; they cannot then be 'moved' from one document to another, for any attempt will be detectable. In digital signature schemes, there are two algorithms: one for signing, in which a secret key is used to process the message (or a hash of the message, or both), and one for verification, in which the matching public key is used with the message to check the validity of the signature. RSA and DSA are two of the most popular digital signature schemes. Digital signatures are central to the operation of public key infrastructures and many network security schemes (eg, SSL/TLS, many VPNs, etc).

Public-key algorithms are most often based on the computational complexity of "hard" problems, often from

number theory. For example, the hardness of RSA is related to the integer factorization problem, while Diffie-Hellman and DSA are related to the discrete logarithm problem. More recently, elliptic curve cryptography has developed in which security is based on number theoretic problems involving elliptic curves. Because of the difficulty of the underlying problems, most public-key algorithms involve operations such as modular multiplication and exponentiation, which are much more computationally expensive than the techniques used in most block ciphers, especially with typical key sizes. As a result, public-key cryptosystems are commonly hybrid cryptosystems, in which a fast high-quality symmetric-key encryption algorithm is used for the message itself, while the relevant symmetric key is sent with the message, but encrypted using a public-key algorithm. Similarly, hybrid signature schemes are often used, in which a cryptographic hash function is computed, and only the resulting hash is digitally signed.

2.4 Authentication

In cryptography, a message authentication code (often MAC) is a short piece of information used to authenticate a message.

A MAC algorithm, sometimes called a keyed (cryptographic) hash function, accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC (sometimes known as a tag). The MAC value protects both a message's data integrity as well as its authenticity, by allowing verifiers (who also possess the secret key) to detect any changes to the message content, and so should be called Message Authentication and Integrity Code: (MAIC).

II. PROJECT WORK

MD5 hash algorithm

Takes as input a message of arbitrary length and produces as output a 128 bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest.

1.Basics of HASH Function

Hash Functions are tools provided by cryptography used to Authenticate and check integrity of message .

A hash value h is generated by a function H of the form

h = H (M)

Where M is a variable-length message and H (M) is the fixed-length hash value.

Hash function is a deterministic procedure that takes arbitrary block of data and

Returns a fixed-size bit string the main purpose of hash function is to produce a "fingerprint" of a message. Hash process is a one-way algorithm it cannot be reversed .

Implementation steps of MD5 Algorithm:

Step 1 – append padded bits:

– The message is padded so that its length is congruent to 448, modulo 512.

• Means extended to just 64 bits shy of being of 512 bits long.

– A single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits equals 448 modulo 512.

Step 2 – append length:

– A 64 bit representation of b is appended to the result of the previous step.

– The resulting message has a length that is an exact multiple of 512 bits.

Step 3 – Initialize MD Buffer

• A four-word buffer (A,B,C,D) is used to compute the message digest.

– Here each of A,B,C,D, is a 32 bit register.

Step 3 cont.

• These registers are initialized to the following values in hexadecimal:

word A: 01 23 45 67

word B: 89 ab cd ef

word C: fe dc ba 98

word D: 76 54 32 10

Step 4 – Process message in 16-word blocks.

– Four auxiliary functions that take as input three 32-bit words and produce as output one 32-bit word.

F(X,Y,Z) = XY v not(X) Z

G(X,Y,Z) = XZ v Y not(Z)

H(X,Y,Z) = X xor Y xor Z

I(X,Y,Z) = Y xor (X v not(Z))

Step 4 – Process message in 16-word blocks cont.

– if the bits of X, Y, and Z are independent and unbiased, the each bit of F(X,Y,Z), G(X,Y,Z),

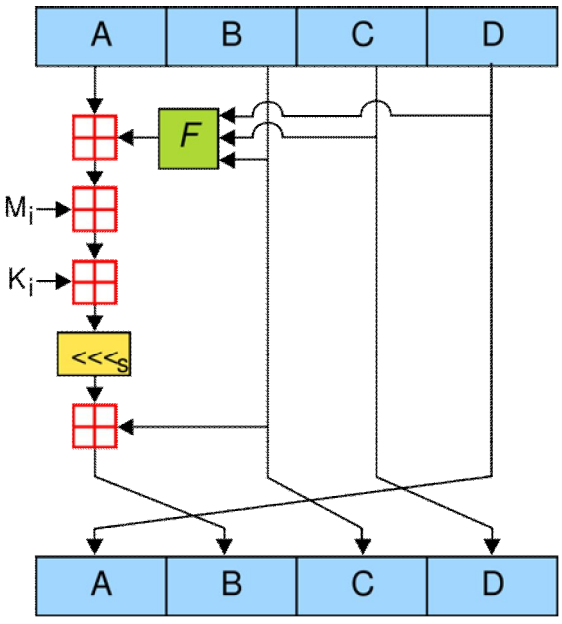
H(X,Y,Z), and I(X,Y,Z) will be independent and unbiased.

Step 5 – output

– The message digest produced as output is

A, B, C, D.

– That is, output begins with the low-order byte of A, and end with the high-order byte of D.



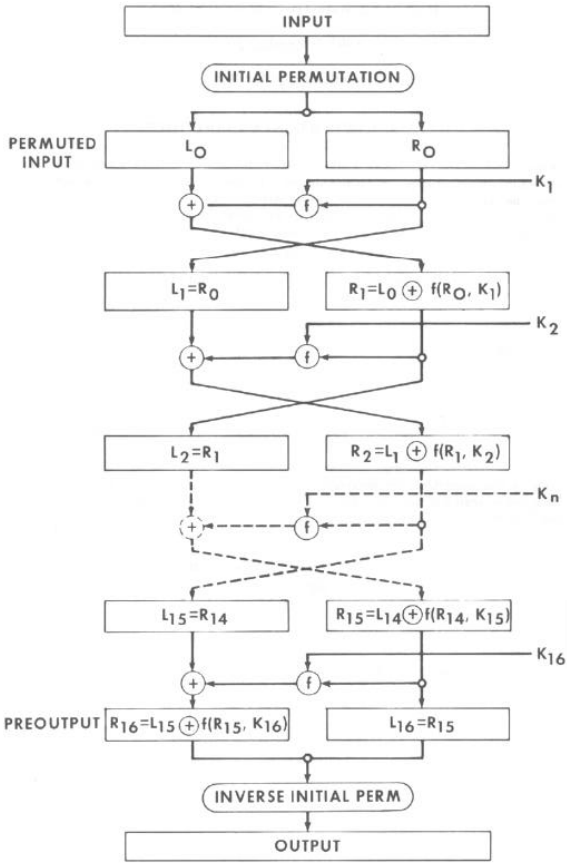
III. DES

The Data Encryption Standard (DES) is a block cipher (a form of shared secret encryption) that was selected by the National Bureau of Standards as an official Federal Information Processing Standard (FIPS) for the United States in 1976 and which has subsequently enjoyed widespread use internationally. It is based on a symmetric-key algorithm that uses a 56-bit key. The algorithm was initially controversial with classified design elements, a relatively short key length, and suspicions about a National Security Agency (NSA) backdoor. DES consequently came under intense academic scrutiny which motivated the modern understanding of block ciphers and their cryptanalysis.

DES is a block cipher, which means that during the encryption process, the plaintext is broken into fixed length blocks and each block is encrypted at the same time. One Block is 64 bits and the key is 64 bits wide (but only 56 bits are used)

The algorithm is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 64-bit key1. Deciphering must be accomplished by using the same key as for enciphering, but with the schedule of addressing the key bits altered so that the deciphering process is the reverse of the enciphering process. A block to be enciphered is subjected to an initial permutation IP, then to a complex key-dependent computation and finally to a permutation which is the inverse of the initial permutation IP-1. The key-dependent computation can be simply defined in terms of a function f, called the cipher function, and a function KS, called the key schedule. A description of the computation is given first, along with details as to how the algorithm is used for decipherment .Next; the use of the algorithm for decipherment is described. Finally, a definition of the cipher Function f is given in terms of primitive functions which are called the selection functions Si and the permutation function P.

The following notation is convenient: Given two blocks L and R of bits, LR denotes the block consisting of the bits of L followed by the bits of R. Since concatenation is associative, B1 B2...B8, for example, denotes the block consisting of the bits of B1 followed by the bits of B2...followed by the bits of B8.



IV. DUAL RSA

Dual RSA:

In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman introduced a cryptographic algorithm, which was essentially to replace the less secure National Bureau of Standards (NBS) algorithm. Most importantly, Dual RSA implements a public-key cryptosystem, as well as digital signatures. Dual RSA is motivated by the published works of Diffie and Hellman from several years before, who described the idea of such an Algorithm, but never truly developed it.

The Dual RSA algorithm is the most commonly used encryption and authentication algorithm and is included as part of the Web browsers from Microsoft and Netscape. It's also part of Lotus notes, Intuit's Quicken, and many other products. The encryption system is owned by RSA Security. The company licenses the algorithm technologies and also sells development kits. The technologies are part of existing or proposed Web, Internet, and computing standards.

DUAL RSA ALGORITHM :

RSA public-key cryptography is the most widely used algorithm in Internet security. Most web browsers implement RSA algorithm to support Secure Socket Layer (SSL) which enables most of today's e-commerce.

In conventional cryptography, a sender and a recipient in secure communication share the same key. The sender uses the key to encrypt a piece of information (plaintext) into an illegible form (ciphertext) and transmit the ciphertext to the recipient over a public network. Only the recipient, who

shares the same key, is able to read the message by decrypting the ciphertext.

The Dual RSA algorithms are described as follows.
Key generation
Encryption
Decryption

Key generation
pick a integer message m
Pick two large prime numbers p and q
Calculate n = p × q
Calculate z= (p − 1)(q − 1)
Pick e also prime number, so that gcd(e, z) = 1,
Range is (1 < e < z)
Calculate d=(e^-1 mod z or (1+z)/e)
Public key (e,z)
Private key (d,z)
Encryption
C=m^e mod n
cipher text(c)
Decryption
Dp =d mod(p-1)
Dq =d mod(q-1)
CP =C^Dp mod p
Cq =C^Dq mod q
M0 =(Cq - CP)p-1 mod q
Decryption message m = CP + M0 p

In practice, the RSA decryption computations are performed in p and q and then combined via the Chinese Remainder Theorem (CRT) to obtain the desired solution in N, instead of directly computing the exponentiation in N. This decreases the computational costs of decryption In two ways. First, computations in p and q are more efficient than the same computations in N since the elements are much smaller. Second, from Lagrange's Theorem, we can replace the private exponent d with dp = d mod (p - 1) for the computation in p and with dq = d mod (q - 1) for the computation in p, which reduce the cost for each exponentiation when d is larger than the primes. It is common to refer to dp and dq as the CRT-exponents. The first method to use the CRT for decryption was proposed by Quisquater and Couvreur. Since the method requires knowledge of p and q, the key generation algorithm needs to be modified to output the private key (d, p, and q) instead of (d, N). Given the private key (d, p, and q) and a valid cipher text C N, the CRT decryption algorithm is as follows:

1) Compute Cp = Cdp mod p.
2) Compute Cq = Cdq mod q.
3) Compute M0 = (Cq - Cp). P-1 mod q.
4) Compute the plaintext M = Cp + M0. P.

This version of CRT-decryption is simply Garner's Algorithm for the Chinese Remainder Theorem applied to RSA. If the key generation algorithm is further modified to output the private key (dp, dq, p, q, p-1 mod q), the computational cost of CRT-decryption is dominated by the modular exponentiations in steps 1) and 2) of the algorithm. When the primes p and q are roughly the same size (i.e., half the size of the modulus), the computational cost for decryption using CRT-decryption (without parallelism) is theoretically 1/4 the cost for decryption using the original method. Using RSA-Small-e along with CRT-decryption allows for extremely fast encryption and decryption that is at most four times faster than standard RSA.

V. DESIGN FLOW

VLSI DESIGN

The complexity of VLSI is being designed and used today makes the manual approach to design impractical. Design automation is the order of the day. With the rapid technological developments in the last two decades, the status of VLSI technology is characterized by the following

• A steady increase in the size and hence the functionality of the ICs.
• A steady reduction in feature size and hence increase in the speed of operation as well as gate or transistor density.
• A steady improvement in the predictability of circuit behavior.
• A steady increase in the variety and size of software tools for VLSI design.

The above developments have resulted in a proliferation of approaches to VLSI design. We briefly describe the procedure of automated design flow the aim is more to bring out the role of a Hardware Description Language (HDL) in the design process. An abstraction based model is the basis of the automated design.

ASIC DESIGN FLOW

As with any other technical activity, development of an ASIC starts with an idea and takes tangible shape through the stages of development as shown in Figure 4.4 and shown in detail in Figure 4.5. The first step in the process is to expand the idea in terms of behavior of the target circuit. Through stages of programming, the same is fully developed into a design description – in terms of well defined standard constructs and conventions.

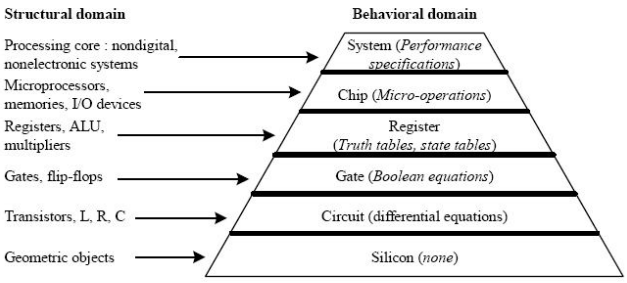


Figure 1.3 Design domain and levels of abstraction.

VI. CONCLUSION

Optimized and Synthesizable VHDL code is developed for the implementation of both encryption process. Each program is tested with some of the sample vectors and output

results are perfect with minimal delay. Therefore, hybrid cryptography algorithm can indeed be implemented with reasonable efficiency on an FPGA. The time varies from chip to chip and the calculated delay time can only be regarded as approximate. Adding data pipelines and some parallel combinational logic in the key scheduler and round calculator can further optimize this design.

ACKNOWLEDGMENT

I am extremely grateful to **Dr. M. Jang Principal** and **Prof M. Guru Nadha Bab Department of ECE, CMR Institute of Tech** their inspiration and valuable guidance during en duration.

First Author Mr. N Krishna Chandu, pursing M.Tech from CMR institute of Engineering & Technology, which is affiliated to JNTU HYD

REFERENCES

[1] V. Klima, "Finding MD5 collisions—A toy for a notebook." Cryptology ePrint Archive. In: CT-RSA. (2004) 324{338 , 2005/075, 2005.

[2] X.Wang, Y. L. Yin, and H. Yu, "Finding collisions in the full SHA-1,"in CRYPTO, V. Shoup, Ed. New York: Springer, 2005, vol. 3621, Lecture Notes in Computer Science, pp. 17–36.

[3] National Institute of Standards and Technology (NIST), MD, "FIPS 180–2, secure hash standard (SHS)," 2002.

[4] Darrel Hankerson, Julio Lopez Hernandez, Alfred Menezes, Software Implementation of Elliptic Curve Cryptography over Binary Fields, 2000, Available at http://citeseer.ist.psu.edu/hankerson00software.html

[5] Certicom, Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, September 2000, Available at http://www.secg.org/download/aid-386/sec2_final.pdf

[6] E. Jochemsz and A. May, "A polynomial time attack on standard RSA with private CRT-exponents", 2007.

[7] M. J. Hinek, "Another look at small RSA exponents," in Topics in Cryptology-CT-RSA 2006, ser. Lecture Notes in Computer Science, D. Pointcheval, Ed. New York: Springer, 2006, vol. 3860, pp. 82–98.

[8] Ravindra Kumar Chahar and et.al., " Design of a new Security Protocol", IEEE International Conference on Computational Intelligence and Multimedia Applications, pp 132 – 134, 2007.

[9] S. D. Galbraith, C. Heneghan, and J. F. McKee, "Tunable balancing of RSA", 2005. Updated version of ACISP 2005.

[10] D. Bleichenbacher and A. May, "New attacks on RSA with small CRTexponent in Public Key Cryptography", PKC 2006, volume 3968 of Lecture Notes in Computer Science, pages 1–13. Springer-Verlag, 2006.

[11] B. den Boer and A. Bosselaers, "Collisions for the compression function of MD5", Advances in Cryptology, Eurocrypt „07, pages 293-304, Springer-Verlag, 2007.

[12] N. Gura, A. Patel, A. Wander, H. Eberle, and S.C. Shantz, "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs". Proceedings of Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004), 6th International Workshop, pages 119–132, 2004.

[13] William Stallings, " Cryptography and Network Security – Principles and Practices", 3rd Edition, Pearson Education Asia – 2003.

[14] Schneier, B., " Applied Cryptography", 2nd Edition, Wiley, 1996.

[15] Rivest, R., " The MD5 message-digest algorithm", RFC 1321, 1992.

[16] D. Johnson, "ECC, Future Resiliency and High Security Systems," Certicom White Paper, March 1999.