Energy Efficient Wireless Encryption

Chetan Nanjunda Mathur and K.P. Subbalakshmi
Media Security, Networking and Communications (MSyNC) Lab,
Department of Electrical and Computer Engineering,
Stevens Institute of Technology,
Hoboken, New Jersey 07030.
Email: {cnanjund,ksubbala}@stevens.edu

***Abstract*— The current encryption standard for wireless networks recommends using the AES cipher in the counter (CTR) mode for confidentiality and the cipher block chaining (CBC) mode for authentication. In the counter mode, a 128 bit counter is encrypted using the AES into 128 bit keystream which is then XORed with 128 bits of plaintext before transmission. This operation is repeated for the entire frame and results in heavy energy consumption for larger frames. In this paper, we propose a novel cipher called High Diffusion (HD) cipher that securely expands a given 128 bit counter value to a larger 288 bit keystream during encryption, thus reducing the number of encryptions per frame compared to the AES. We show that the HD cipher is as secure as the AES under differential, linear cryptanalysis and Square attack. Using an experimental set up consisting of a laptop with 1.8 GHz Pentium 4 processor and an Intrinsyc CerfCube with 233 MHz ARM processor we measure the energy consumption of both the AES and the HD cipher encryption operation. We observe that using HD cipher instead of AES for encryption will result in about 40% saving in energy consumption on both the laptop and the CerfCube. When HD cipher is used instead of AES in the CCMP, we observe that energy efficiency due to HD cipher is significant for larger frame lengths.**

I. Introduction

The current security mechanisms for the 802.11 wireless LAN standards are specified in the amendment IEEE 802.11i [3], also known as Wi-Fi Protected Access 2 (WPA2). The WPA2 makes use of the Advanced Encryption Standard (AES) [10] block cipher (based on Rijndael [8]); to provide both authentication and confidentiality in a single protocol called the Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP). The 802.11i architecture [3] contains the following components: 802.1X for authentication (entailing the use of Encapsulated Authentication Protocol (EAP) and an authentication server), Robust Security Network (RSN) for keeping track of associations, and AES based CCMP to provide confidentiality, integrity and origin authentication.

The reason for the switch from the RC4 based WEP [17] and TKIP [18] (the predecessors to WPA2) to the AES based CCMP was due to the superior security of the AES in comparison to the RC4. However, the drawback of AES-CCMP is that, it consumes more energy compared to its predecessor. This is because, the RC4 cipher used in WEP is a stream cipher; whereas, the AES used in CCMP is inherently a block cipher used in stream (counter or CTR) mode. *Therefore, a full 10 round AES needs to be performed to encrypt every 128 bits of MPDU*. For larger frame sizes (or data payload in MPDU), this approach is energy inefficient.

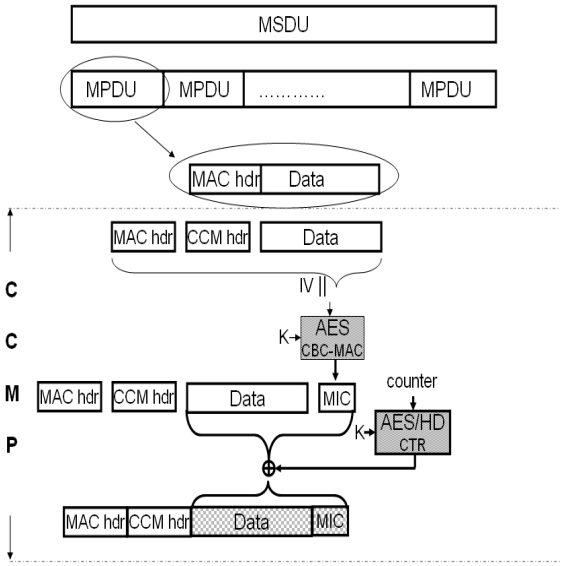

Fig. 1. Block Diagram of CCMP.

In this paper, we address this problem by using a novel cipher called High Diffusion (HD) cipher. The proposed HD cipher is similar in structure to AES with one important difference, in that, *the HD cipher can securely encrypt k bit input data to n bit encrypted output data (where $k < n$). This secure expansion property when used in the CTR mode results in higher encryption throughput*. For example, with the appropriate choice of parameters, one HD cipher encryption in the CTR mode can encrypt 288 bits of information as opposed to 128 bits using the AES. Although a single encryption of HD cipher is only slightly more computationally expensive compared to a single encryption of the AES, the HD cipher requires only half the number of encryptions to encrypt the entire MPDU.

The rest of the paper is organized as follows, in Section II we briefly introduce the CCMP and describe some of its drawbacks. We then propose to improve the energy efficiency of CCMP using a novel HD cipher in Section III. In Section IV we analyze the security of HD cipher against most of the well known attacks. Experimental results comparing energy

consumption of our proposed HD-CCMP against AES-CCMP is given in Section V followed by Conclusions in Section VI.

II. Counter mode CBC-MAC Protocol (CCMP)

The CCMP is executed at the link layer of 802.11 wireless devices, where the input data called the message service data unit (MSDU) is often broken down into many message protocol data units (MPDUs), where each MPDU consists of MAC header and a data payload. Each of the MPDU's are processed using CCMP to produce encrypted MPDU's with tagged Message Integrity Check (MIC) value and cleartext headers. Fig. 1 describes the flow of data through the CCMP. The CCMP requires two state variables: a single shared key (K) for both encryption and authentication and a 48-bit packet sequence number (PN). The CCMP uses the PN to construct both the counter for encryption and the initialization vector (IV) for authentication. A 64-bit message integrity check (MIC) value is generated by encrypting the IV, the entire MPDU and the CCMP header with 128-bit AES in CBC-MAC mode, which is then appended to the MPDU. The data payload of the MPDU and the MIC are encrypted using the 128-bit AES in the CTR mode. The value of the counter is incremented after every 128-bit block encryption. At the receiver, the encrypted MPDU's are decrypted using the PN extracted from the cleartext headers and the shared secret key. The decrypted MPDU's are then authenticated against the tagged MIC values. A detailed description of the protocol including the construction of counter, IV etc can be found in [3].

The CCMP is secure as long as: (a) a (key, IV) pair is not reused (b) a (key, counter) pair is not reused. A new shared secret key is established for every session. The MSDU for that session may be broken down to many MPDUs, each MPDU is assigned a unique 48 bit PN starting with zero and incremented for every additional MPDU, to ensure that there is no repetition within a given session. Additionally, the inclusion of MAC address in the IV prevents the sender and the receiver from using the same (key, IV) pair. Similarly, the counter value begins with zero and increments for every 128 bit block of data payload to be encrypted. The per-packet block counter is 16 bits, therefore we can uniquely encrypt up to 65536 blocks or (1048576 bytes). This easily accommodates the largest MPDU (2312 bytes) allowed in IEEE 802.11.

III. Designing energy efficient CCMP

One way to improve the energy efficiency of the CCMP, is to introduce a cipher that can encipher larger chunks of the plaintext per encryption. That is, we need a high encryption throughput cipher that replaces the 128-bit AES used in the CCMP. We define the term *encryption throughput* of a cipher as the number of plaintext bits encrypted with one single encryption of the cipher. For example, the 128-bit AES used in CCMP has an encryption throughput of 128 bits/encrytion. In order to achieve higher encryption throughput, we could use an AES which operates on 256-bit blocks of input data to produce 256-bit key stream. However, such an approach would not be energy efficient as the 256-bit AES consumes significantly more energy per encryption compared to the 128-bit AES.

In this paper, we take a novel approach to increase the encryption throughput by introducing a diffusion function that also causes expansion. This can be done by using a new class of codes that we call the high diffusion code (HD-codes) [13] at the diffusion layer. Although, these codes were designed with the specific goal of constructing error correcting ciphers [14], these can also be used for our purposes since the codes provide a means of securely increasing the encryption throughput. Note that, in [14] we first introduced the HD-cipher in the electronic codebook (ECB) mode and present a proof-of-concept construction using a specific construction in GF(8). In this mode (and the cipher block chaining mode) the primary function of the HD-cipher is to correct transmission errors. In this work, we go beyond our previous work, by (a) constructing a full (GF(256)) construction of the cipher and presenting its use in the CTR mode (b) providing experimental results on "real life" power constrained devices like laptops and the Intrinsyc CerfCube to demonstrate the viability of this approach.

A. The High Diffusion Cipher in GF(256)

The HD cipher [14] is a key-alternating [6] block cipher, composed of several iterations of the round transformation and key mixing operation. The round transformation consists of three layers: a) the non linear substitution layer, b) symbol transposition layer and c) the High Diffusion encoding layer. The HD encoding layer takes in k_r bits of input and produces n_r bits of output, where $k_r < n_r$ and at each round r. Note that, the HD encoding is not performed in the final round. The key mixing layer follows every round transformation and is also performed once before the first round. The input data, as it goes through each round of the cipher, is referred to as the *cipher state*.

The round keys are generated using the key expansion algorithm, which is similar to that of the AES key expansion algorithm [10], to construct 11 round keys. All the operations in HD cipher are performed in the finite field of order 2^8, denoted by GF(256). A detailed description of all the layers of HD cipher follows:

1) Key Mixing Layer: The key mixing layer is a bitwise XOR operation of the cipher state with the round key. Note that, the output cipher state of the key mixing layer of round $r-1$ forms the input cipher state to the next round r. However, when $r = 10$, the output cipher state is the ciphertext (keystream in CTR mode).

2) Non-linear Substitution Layer: The substitution layer uses an invertible local non-linear transformation called the S-box. The non-linearity in S-box is desired to cause intra symbol avalanche [9] (that is every bit in the output symbol of the S-box flips with a probability of half for a single bit flip in the input symbol), which is essential for the security of the cipher. Nyberg proved that substitution functions generated by inverting elements in GF(2^8) are differentially 4 uniform

and are highly nonlinear [16]. The S-boxes thus constructed are used in the substitution layer of the HD cipher. Note that, these S-boxes are also used in the substitution layer of the AES.

3) Symbol Transposition Layer: The symbol transposition layer the is the first of the two diffusion operations used in the HD cipher. The aim of this layer is to permute the cipher state using a diffusion optimal transformation. We use the matrix transpose operation which was shown to be diffusion optimal [14].

4) HD Encoding Layer: The HD encoding transformation is the second diffusion operations used in the HD cipher. The aim of this layer is to diffuse the intra symbol avalanche caused by the substitution layer to a large number of symbols in the resulting cipher state. In our application (HD-cipher in the stream mode), this layer has an additional aim, which is to securely expand the input cipher state to a larger output cipher state. The HD encoding operation uses novel channel codes called High Diffusion codes [13] which posses the high diffusion property or spreading strength (measured in terms of the branch number [8]) and are defined as follows:

Definition 1: High Diffusion codes are $[n, k, q]$ Maximum Distance Separable (MDS) codes with branch number equal to $n + 1$.
This is the maximum possible branch number any function can possess.

By picking appropriate parameters for the HD code, it is possible to achieve the desirable level of expansion in the diffusion layer and therefore an appropriate amount of savings in energy consumption. In our experiments, we construct a 10 round HD-cipher with input data size of 128 bits and output ciphertext (or keystream in CTR mode) and keysize of 288 bits. This is achieved by using a [4,4,256] HD code for rounds 1 through 7 and a [6,4,256] HD code for rounds 8 and 9. The generator matrixes for these HD codes are,

$$G(r)_{r=[1...7]} = \begin{pmatrix} 1 & 1 & 3 & 2 \\ 2 & 1 & 1 & 3 \\ 3 & 2 & 1 & 1 \\ 1 & 3 & 2 & 1 \end{pmatrix}$$

$$G(r)_{r=[8,9]} = \begin{pmatrix} 1 & 1 & 3 & 2 & 189 & 71 \\ 2 & 1 & 1 & 3 & 169 & 27 \\ 3 & 2 & 1 & 1 & 192 & 209 \\ 1 & 3 & 2 & 1 & 91 & 179 \end{pmatrix}$$

To perform HD encoding, each column of the input cipher state is multiplied with $G(r)$ to obtain the output cipher state. The branch number $\mathcal{B}(G(r))$ of $G(r)_{r=[1...7]}$ is 5 and $G(r)_{r=[8,9]}$ is 7.

B. Using HD cipher in CCMP

By using the HD cipher in the CTR mode for confidentiality instead of the AES (see Fig. 1) we propose to make CCMP more energy efficient. The construction of the counter from the PN remains the same as in CCMP [3]. Let C denote the 128 bit counter, X denote the MPDU stream (data payload + MIC) of length N-bits to be encrypted and X_i denote the i-th 288 bit block of X (where, $i \in \{1...\lceil N/288 \rceil\}$). The HD encryption under key K, denoted by E_K, in the CTR mode is $Y_i = E_K(\text{counter} + (i-1)) \oplus X_i,\ \forall i \in \{1...\lceil N/288 \rceil\}$. Here, Y represents the encrypted MPDU stream. In the CTR mode, the HD cipher securely expands the 128 bit counter to 288 bit keystream, thus encrypting more than twice the number of bits per encryption compared to AES. The number of encryptions required by the HD cipher per N bit frame is $\lceil N/288 \rceil$, whereas for the AES it is $\lceil N/128 \rceil$. We therefore expect to do only 50% of work to achieve the same level of confidentiality compared to the AES-CCMP. Moreover, as n increases we expect to observer larger reduction in energy consumption. Although, from a security standpoint we can use the HD cipher in CBC-MAC mode for authentication as well, this does not cause additional savings in terms of energy, hence we use the HD-cipher in the CTR mode for confidentiality only. From an implementation standpoint, most of the operations of the HD cipher are similar to the AES and significant portions of code can be reused. Hence, using two different ciphers in CCMP does not impose a significantly larger code space requirement.

IV. Security Analysis of HD ciphers

In this section, we analyze the security of HD ciphers by looking at the resistance it offers against some well known cryptanalytic attacks.

Differential cryptanalysis [4], [5] is a chosen plaintext-ciphertext attack that makes use of difference propagation property of a cipher to deduce the key bits. The difference propagation property of an S-box is the relative number of all input pairs, that for the given input difference, give rise to a specific output difference. It is expressed as propagation ratio [6]. Let $\vec{x}_1^r$ be any intermediate cipher state at round r resulting from the input P_1. Similarly, let $\vec{x}_2^r$ be the corresponding intermediate cipher state resulting from P_2. The non zero symbols in $\vec{x}_1^r \oplus \vec{x}_2^r$ are called active symbols or S-boxes. The difference propagation of consecutive round can be concatenated across several rounds to form a differential trail. The propagation ratio over all the rounds of a differential trail can be approximated by the product of the propagation ratios of its active S-boxes. Differential cryptanalysis can break the HD cipher with complexity less than $\mathcal{O}(2^{128})$ if the maximum possible propagation ratio over all rounds is significantly larger than 2^{-127}.

Linear cryptanalysis [15] is a known plaintext-ciphertext attack that makes use of linearity in the cipher to obtain the key bits. The substitution is the only non-linear step in most of the block ciphers including the proposed HD cipher. The linearity of an active S-box can be approximated to the maximum input-output correlation exhibited by it. The active S-boxes in a round are determined by the non zero symbols in the selection vectors at the input of the round. The linearity of one round can be extended to multiple rounds to form a linear trail. The correlation (measure of linearity) of a linear trail

(multiple rounds) can be approximated to the product of input-output correlations of its active S-boxes. Linear cryptanalysis can break the HD cipher with complexity less than $\mathcal{O}(2^{128})$ if the maximum possible correlation of any linear trail over all rounds is significantly larger than 2^{-64}.

Hence, a lower bound on the number of active symbols in any linear or differential trail will give a lower bound on the resistance of the cipher to linear and differential cryptanalysis. In [14] we show that this lower bound for any four rounds of HD cipher, starting with round r is $\mathcal{B}(G(r)) \times \mathcal{B}(G(r+1))$. The lower bound on the number of active S-boxes in any linear or differential trail in the last four rounds of the HD cipher proposed here is $\mathcal{B}(G(7))\mathcal{B}(G(8))$ or 35. The S-boxes used in the substitution layer of HD cipher have a maximum propagation ratio of 2^{-6} and maximum input and output correlation of 2^{-3}. This shows that there are no four round differential trails with predicted propagation ratio above 2^{-215} and no four round linear trails with predictable input output correlation above 2^{-105}. The initial six rounds are added as a security margin towards future attacks, just as in AES. Hence the 10 round 128 bit HD cipher is secure against linear and differential cryptanalysis.

The Square attack [7] (also known as Integral attack or the Saturation attack) makes use of the byte oriented nature of the Square block cipher which was the predecessor of AES. As AES is also a byte oriented cipher, this attack has been extended to reduced versions of AES [11], [12]. The proposed HD cipher also comprises of byte oriented operations which are loosely based on AES, hence HD ciphers with fewer than seven rounds would be as weak as reduced versions of the AES.

Although the HD cipher is as secure as AES against most of the well known attacks, the HD cipher uses a larger key length to achieve the same security level as that of AES. Since, the key expansion is performed only once every session, its computational overhead is negligible.

V. Experimental Results

Two sets of experiments were conducted, one on a laptop and one on the Intrinsyc CerfCube [2]. The CerfCube represents an environment with severe battery power and computational resource constraints and the laptop represents an environment in which the resource constraints can be relaxed. The testbed (Fig 2) consists of a Sony Vaio laptop with a 1.8 GHz intel P-4 processor, 512 MB RAM, running Red Hat Linux 2.4.8 and a Intrinsyc CerfCube [2] with a 233 MHz ARM processor, 16MB Flash and 32 MB SDRAM, running Debian linux operating system. The power consumed by the CPU in running the encryption algorithms is measured as a function of input power supply to the laptop/CerfCube. A separate DC power supply is given to the laptop/CerfCube to permit measurements. The battery of the laptop is removed for accuracy in the measurements. The current is measured using Labview from the GPIB interface of the power supply. To eliminate effects of any programs running in the background, the current consumption is first tested when no other tasks

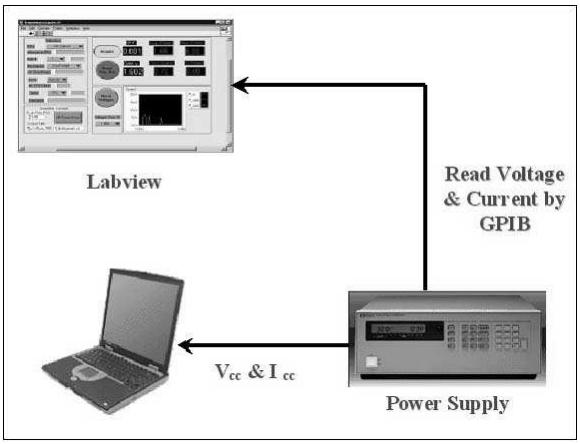

Fig. 2. Hardware Setup

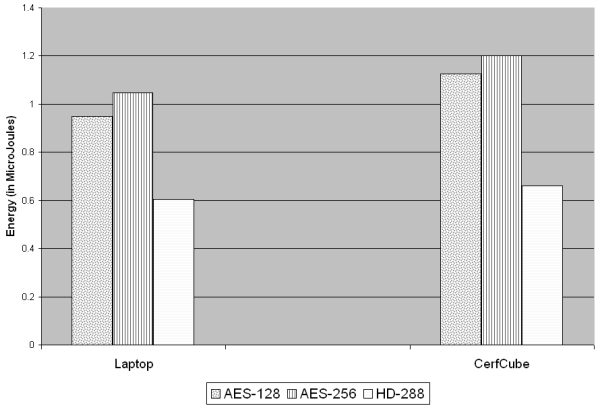

Fig. 3. Energy consumed per byte due to 128 bit AES, 256 bit AES and 288 bit HD encryption on laptop and Intrinsyc CerfCube

are running. The difference in currents when the algorithm is running and the idle current (in Amperes) is taken as the actual current consumption. In the experiments, since voltage variation is seen to be extremely small (measured to be less than 0.025%) we use a constant value. We use OProfile [1] to measure the exact time taken by the algorithms to run. The energy consumed by the algorithms is the product of power drawn from the DC source and the time required to complete execution.

We measured the energy consumed by a full 10 round 288-bit (keystream length) HD cipher, 10 round 128-bit AES and a 256-bit AES on both the CerfCube and the laptop. The measured energy consumption is divided by 36, 16 and 32 respectively to obtain the per byte energy consumption. Figure. 3 plots the per byte energy consumption due to AES and HD cipher. It can be observed from the figure that HD cipher results in about 40% reduction in energy consumption in both the CerfCube and the laptop. We then measured the energy consumption per frame due to AES and HD cipher for various data payload lengths of MPDU. Fig. 4 plots the per frame

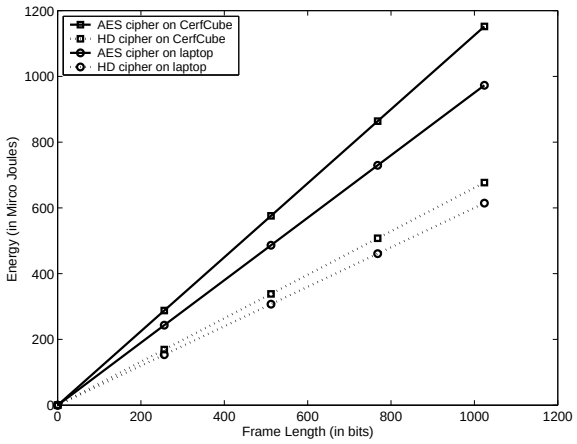

Fig. 4. Energy consumed per frame due to 128-bit AES (denoted by solid line) and HD encryption (denoted by dotted line) on the CerfCube (denoted by circle) and laptop (dented by square)

energy consumption by 128-bit AES and the proposed HD cipher used in CCMP. We observe that HD cipher consumes significantly less energy compared to the AES as the frame length gets larger. Another interesting observation is that, both AES and HD cipher consume slightly more energy in the CerfCube compared to the laptop. This indicates that as we move to more resource constrained environments, the use of our proposed energy efficient protocol is warranted.

VI. Conclusions

In this paper, we propose a HD cipher that securely encrypts a 128 bit counter value to a larger (for example 288 bit) keystream. Replacing the AES used in CCMP with the HD cipher allows us to achieve higher encryption throughput. Energy analysis experiments reveal that HD cipher consumes 40% less energy compared to the traditional AES-CCMP. Also, the proposed system performs significantly better when larger frame lengths are used, thereby demonstrating the significant energy gains that could be achieved in resource constrained systems.

VII. Acknowledgement

This work was partially supported by NSF Cyber Trust Grant No.0627688 and US Army/Picattinny Arsenal.

References

[1] http://oprofile.sourceforge.net.
[2] http://www.intrinsyc.com/products/mob_ref_sys/cerfcube_255/.
[3] Amendment 6: Medium access control (mac) security enhancements. *802.11i, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, July 2004.
[4] E. Biham and A. Shamir. Differential cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer (extended abstract). *Lecture Notes in Computer Science*, 576:156, 1991.
[5] E. Biham and A. Shamir. Differential cryptanalysis of the full 16-round des. In *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pages 487–496, London, UK, 1993. Springer-Verlag.
[6] J. Daemen. *Cipher and hash function design strategies based on linear and differential cryptanalysis*. PhD thesis, K.U.Leuven, March 1995.
[7] J. Daemen, L. R. Knudsen, and V. Rijmen. The block cipher square. In *FSE '97: Proceedings of the 4th International Workshop on Fast Software Encryption*, pages 149–165, London, UK, 1997. Springer-Verlag.
[8] J. Daemen and V. Rijmen. *The design of Rijndael: AES — the Advanced Encryption Standard*. Springer-Verlag, 2002.
[9] H. Feistel. Cryptography and computer privacy. 228(5):15–23, May 1973.
[10] FIPS. Specification for the advanced encryption standard (AES). Federal Information Processing Standards Publication 197, 2001.
[11] H. Gilbert and M. Minier. A collision attack on 7 rounds of rijndael. In *AES Candidate Conference*, pages 230–241, 2000.
[12] S. Lucks. Attacking seven rounds of rijndael under 192-bit and 256-bit keys. In *AES Candidate Conference*, pages 215–229, 2000.
[13] C. N. Mathur, K. Narayan, and K. Subbalakshmi. High diffusion codes: A class of maximum distance separable codes for error resilient block ciphers. *2nd IEEE International Workshop on Adaptive Wireless Networks (AWiN), Globecom*, November 2005.
[14] C. N. Mathur, K. Narayan, and K. Subbalakshmi. High diffusion cipher: Encryption and error correction in a single cryptographic primitive. To appear in the 4th International Conference on Applied Cryptography and Network Security Conference (ACNS), June 2006.
[15] M. Matsui. Linear cryptanalysis method for DES cipher. In *Advances in cryptology -EUROCRYPT93, Lecture Notes in Computer Science*, volume 765, pages 1–11, 1993.
[16] K. Nyberg. Differentially uniform mappings for cryptography. In *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 55–64, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.
[17] L. M. S. C. of the IEEE Computer Society. Wireless lan medium access control (mac) and physical layer (phy) specifications. 1999.
[18] J. Walker. 802.11 security series part ii: The temporal key integrity protocol (tkip). *Technical report, Platform Networking Group, Intel Corporation*.