

# Wireless Network Security for IEEE 802.11 Infrastructure

Sajeet Giga, Neco Ventura, H. Anthony Chan  
Department of Electrical Engineering, University of Cape Town,  
Rondebosch, Cape Town, South Africa

{sgiga, neco, achan}@crg.ee.uct.ac.za

**Abstract**— The main goal of this paper is to collate and independently apprise the large body of works that deal with the security of the IEEE 802.11 Standard. The paper reviews the Wired Equivalent Privacy (WEP) protocol and confirms its insecurity. It also includes an evaluation of the recently ratified IEEE 802.11i Amendment and a performance assessment study of contemporary 802.11 security techniques

**Index Terms**—IEEE 802.11, Wireless Networks, Security Evaluation, Network Performance

## I. INTRODUCTION

The widespread deployment of Wireless Local Area Networks (WLANs) is due in large part to the presence of the IEEE 802.11 Standard and its leverage of the unlicensed Industrial, Scientific and Medical (ISM) frequency spectrum. The 2.4 GHz to 2.4835 GHz UHF band and the 5.15 GHz to 5.825 GHz SHF band are both used for this purpose.

The IEEE 802.11 working group was established in 1990 and the original 802.11 Standard was subsequently ratified in 1997. The 802.11 Standard addresses the lower levels of the Open System Interconnection (OSI) Reference Model by independently defining both the Physical and Data-link layers. Enhancements to the original standard have since been developed and these include 802.11b, 802.11a and 802.11g.

The convenience offered by wireless networks comes at the price of heightened security concerns, caused by the very nature of the transmission medium used. The media frenzy regarding the inability of the Wired Equivalent Privacy (WEP) protocol to properly secure wireless transmissions has dampened the proliferation of IEEE 802.11-based technology, especially in the corporate environment.

Although these security concerns are somewhat justified, many myths regarding the security of wireless networks abound. This paper attempts to provide an independent evaluation of recently released 802.11 security mechanisms and examines their ability to effectively address these concerns.

## II. WIRELESS EQUIVALENT PRIVACY (WEP)

The original 802.11b Standard for WLANs introduced the Wired Equivalent Privacy (WEP) protocol in an attempt to address the heightened security concerns regarding the use of wireless networks. The main goal of the protocol was to bring the security of such networks closer to that of generic wired infrastructure. This was to be achieved by assuring the confidentiality of user data in the event of eavesdropping over the radio frequency interface.

The WEP protocol is used in 802.11-based networks to protect Data-link layer data during wireless transmission. The WEP protocol is defined to be self-synchronizing for each data-link frame. This property is critical for wireless protocols as ‘best effort’ transmission is often assumed due to high bit error rates. The WEP protocol is also designed to impose as little overhead as possible, with hardware implementations of all algorithms wherever possible.

The remainder of this section attempts to provide an overview of WEP by describing the main features of the protocol, with emphasis placed on security concerns.

### A. Authentication

The 802.11 Standard specifies two authentication algorithms, open network authentication and shared key authentication. Manufacturers and network administrators have also implemented authentication mechanisms based on the use of access control lists and a form of security through obscurity known as closed network authentication.

Open Authentication is a null authentication algorithm as any station requesting authentication is granted access. All the station requires is knowledge of the Service Set Identifier (SSID) of an access point. Beacon Management frames periodically broadcast this and other network information.

Shared Key Authentication utilises a standard challenge and response dialog together with a symmetric, shared key to provided authentication. Mutual authentication does not occur and the entire process is dependent on the security of the WEP encryption key. This leaves 802.11 networks susceptible to ‘Man in the Middle’ attacks, in which an attacker makes use of ‘rogue’ access points.

Additional security can be gained by restricting network access to known adapters on an authorized MAC access control list [1]. In theory, access control lists provide a reasonable level of security when a strong form of identity is used. However, this property is not possessed by MAC addresses which are easily spoofed and are transmitted over the airwaves in an unencrypted form.

**This work was supported in part by Telkom SA, Siemens, the National Research Foundation (NRF) and the Department of Trade and Industry (DTI).**

Access points implementing Closed Network Authentication do not respond to Access Probe Request management frames and omit the network SSID from Beacon Frames. As a result, stations cannot detect the presence of the 802.11 network unless the SSID is obtained manually [13]. Unfortunately, SSID information is still transmitted unencrypted in the headers of Association Request management frames, from stations with prior knowledge of the SSID. As a result, the SSID of an access point can be determined by monitoring the 802.11 frequency spectrum with appropriate tools.

### B. Integrity

The Cyclic Redundancy Check (CRC-32) algorithm is used to derive an Integrity Check Value (ICV) for the data payload of each and every 802.11 frame. The CRC-32 checksum was originally developed to detect random errors, induced during transmission of a given block of data, and was not intended to be used as a security mechanism. Unfortunately, the linear nature of the algorithm allows controlled modifications to be made to a cipher text without violating the checksum [2].

This weakness allows arbitrary bit flipping modifications to be performed on an encrypted message without any fear of detection. This process is further simplified by the fact that the CRC checksum is an un-keyed function of the message that it is protecting.

### C. Encryption Algorithm

The WEP encryption algorithm is based on the RC4 stream cipher. The RC4 encryption algorithm is a symmetric stream cipher that supports a variable-length key. RC4 allows the key length to be up to 256 bytes in size; however the IEEE originally specified that 802.11 devices support 64-bit keys. In recent times, 128-bit, 152-bit and 256-bit keys have been introduced.

The WEP protocol uses the RC4 stream cipher in an environment for which it is not suitable. For data encrypted with a synchronous stream cipher, loss of a single bit of the data-stream causes the loss of all subsequent data. This is due to the fact that any data loss de-synchronizes the key-stream generators at both end-points of the transmission. Since wireless networks are especially susceptible to data loss, it is infeasible to use a synchronous stream cipher across 802.11 frame boundaries. The cipher must thus be initialized for each frame requiring encryption, exposing the cipher to a number of vulnerabilities.

A well known danger regarding the use stream ciphers is that encrypting two messages using the same key-stream can reveal information regarding both messages [2]. To negate this vulnerability, WEP uses per-frame Initialization Vectors (IVs) to vary the seed used by the key-stream generation process. The IV is transmitted in the unencrypted portion of each WEP frame so that the receiver is able to determine the correct decryption key.

The WEP standard recommends that the IV be changed for every frame. Most implementations simply increment the IV field after every frame and reset the IV field when the 802.11 adapter is initialized. This can lead to low valued IVs

being used more often than other IV values. Furthermore, the IV field used by WEP is only 24 bits long – ensuring that the IV space will be quickly exhausted for moderate transmission rates. For example, transmission of 1540 byte frames at half the capacity of the network (5.5 Mbps) will exhaust the IV space in just under 10½ hours.

Over time, a full table of all possible IVs and corresponding key-streams can be compiled. Assuming 1540 bytes for a key-stream and  $2^{24}$  possible IVs, this table has storage requirements of approximately 24 GB [2]. Once the table is complete, very little effort is required to decrypt each and every cipher text transmitted across the network.

The most important weakness of the WEP protocol is based on the partial key exposure attack against the RC4 algorithm, discovered by Fluhrer et al in 2001 [5] and implemented and refined by both Stubblefield et al [12] and Hulton [9] in 2002.

Given many prefixes of the seed to the RC4 algorithm, the Fluhrer, Mantin and Shamir (FMS) attack is able to recover the next byte of the seed. By iterating the process, the full seed can be thus be recovered. In the WEP implementation of RC4, the seed is the concatenation of the Initialization Vector (IV) and the shared WEP key, in that order. As the IV is transmitted unencrypted, the required prefix is easily acquired and since the shared WEP key is seldom changed, the rest of the seed remains constant.

The FMS attack relies on an observation of only the first byte of the pseudo-random sequence generated by the RC4 algorithm. This byte is itself dependent on only three values of the RC4 state permutation. Certain IVs initialise this state in such a way that leaks information about the rest of the seed. This information can then be used to correctly guess the next key byte, with an accuracy of 5%, provided that the previous bytes of the key are known [5]. Fluhrer et al. [5] estimate that approximately 60 resolving IVs are theoretically required to recover the full WEP key and that the attack requires approximately 4,000,000 frames to achieve this.

Stubblefield et al [12] provide an implementation of the FMS attack and postulate optimizations to the attack. These optimizations include an increased set of candidate resolving IVs, use of resolving IVs for late key bytes to improve the guess of early key bytes, and the exploit of other special resolving cases. These optimizations are able to reduce the number of frames required to recover a 104-bit WEP key to approximately 1,000,000 frames.

Hulton [9] provides further enhancements to the FMS attack. Further attacks on bytes other than the first RC4 output byte, together with a degree of brute-force and ‘fudging’ techniques, are proposed and an implementation is presented. The result of these enhancements is an efficient FMS attack capable of successfully recovering a WEP key with as little as 500,000 frames and less than 60 weak IVs.

## III. IEEE 802.1X

In response to the insecurity of authentication mechanisms provided by the WEP-secured 802.11 Standard, the IEEE proposed a new long-term security architecture known as the Robust Security Network (RSN). This architecture is based

on the IEEE 802.1X Standard, which provides access control, authentication and key management services [11].

The 802.1X standard specifies a generic method for the provision of port-based access control to the Data-link layer of the OSI Reference Model. Although designed to operate on top of a number of different networks such as 802.3 Ethernet, the 802.1X standard does however include a number of features aimed specifically at improving the security of 802.11-based networks.

The 802.1X Standard is based on the Extensible Authentication Protocol (EAP). EAP is a generalized framework that facilitates the operation of a number of different authentication mechanisms between users and an authentication server, at the network layer of the OSI model. It effectively creates a tunnel over the underlying network structure to the authentication server, thereby allowing the use of more powerful authentication protocols.

The 802.1X framework requires the presence of three entities - the supplicant, the authenticator and an authentication server. These entities are logical, software-based entities running on network hardware devices.

The supplicant resides on each wireless station that requires access to network services. These services are available only through an authenticator. The supplicant acts on behalf of the wireless station and is responsible for supplying the station's credentials to the authenticator on request.

The authenticator resides on the access point and enforces authentication before authorizing supplicant access to services through the logical port. It is thus responsible for obtaining supplicant credentials, which are then passed on to an authentication server. The authenticator is necessary because the supplicant has no knowledge of where to find the authentication server, since it is only able to send and receive Data-link traffic.

The authentication server is responsible for verifying the supplicant credentials that are passed to it by the authenticator. Notification regarding the success or failure of the authentication procedure is passed back to the authenticator which can then take appropriate actions.

EAP-TLS (Transport Layer Security) is an authentication method based on the Secure Socket Layer (SSL) protocol [10]. EAP-TLS offers an extremely secure authentication method that is based on the use of Public Key Infrastructure (PKI) digital certificates for both the authentication server and supplicants. Mutual authentication involves the secure exchange of these certificates and thus mitigates the risk of 'Man in the Middle' attacks.

EAP-TTLS (Tunnelled Transport Layered Security) was developed to provide similar security to EAP-TLS, without the burden of PKI certificate management [6]. EAP-TTLS, like EAP-TLS, is based on the SSL protocol but uses certificates to authenticate the authentication server only. An encrypted tunnel is established, using the SSL Record protocol, through which authentication of the supplicant takes place. This allows supplicant authentication to be carried out using much simpler mechanisms, which would otherwise be totally insecure for use in a wireless network environment.

EAP-PEAP (Protected EAP) is a tunneling EAP

mechanism that is very similar to EAP-TTLS. As with EAP-TTLS, an asymmetric encryption tunnel is created using server certificate-based authentication, through which secure client authentication then occurs. Unlike EAP-TTLS, PEAP does not support legacy supplicant authentication methods and requires that the inner authentication method be based on EAP itself [6]. PEAP is thus more suited to new network deployments that do not need to support legacy infrastructure. Furthermore, PEAP has been adopted as the EAP method of choice by Microsoft and is supported by the latest versions of Windows.

#### IV. IEEE 802.11i AMENDMENT

Although the IEEE 802.11 Standard attempted to address security concerns through the introduction of the Wired Equivalent Privacy (WEP) protocol, WEP proved relatively insecure for the wireless network paradigm. In response, the IEEE formed the 802.11i Task Group whose three and a half years of work culminated in an amendment adding stronger encryption, authentication and key management strategies to the 802.11 Standard.

The IEEE 802.11i amendment has numerous innovative components, the most important of which include the two new data confidentiality protocols TKIP and CCMP. IEEE 802.11i also leverages the IEEE 802.1X authentication and key-distribution mechanisms and separates the handling of unicast and broadcast traffic. The use of pre-shared keys is supported for small networks in which IEEE 802.1X deployment is not justified. A new negotiation process between network entities is also provided to select between the different confidentiality mechanisms on offer. Other novel features include support for key-caching and pre-authentication [7].

##### A. TKIP

The Temporal Key Integrity Protocol (TKIP) is designed to mitigate all the vulnerabilities of the Wired Equivalent Privacy (WEP) protocol while maintaining backward-compatibility with 802.11 hardware already deployed. TKIP was thus designed to be implemented through simple firmware or driver upgrades to existing 802.11-based infrastructure and to minimize performance degradation imposed by the fixes. TKIP provides two major enhancements to WEP: a Message Integrity Code (MIC) function and per-packet keying for all WEP-encrypted frames.

Conventional MIC hash functions were found to be too computationally expensive to execute on existing 802.11 hardware and a new MIC called 'Michael' was thus defined [4]. The instruction set used to perform the hash is deliberately limited to preserve backward-compatibility and reduce performance degradation. The Michael algorithm only provides 20-bits of security, however, and thus TKIP defines additional security measures. TKIP requires a re-key after every MIC validation error and limits re-keying to once every minute. This limits the effect of brute force, birthday and differential cryptanalytic attacks [8].

The original 802.11 standard did not specify a means for the distribution of WEP keys and most attacks on the WEP protocol thus rely on the presence of a static WEP key. Use

of per session keying via the 802.1X re-authentication mechanism mitigates some of these vulnerabilities but does not provide complete protection against the inherent weakness present in WEP's use of the RC4 stream cipher.

TKIP thus introduces a per-frame encryption key construction based on a mixing function. This mixing function takes a number of inputs, which include a 128-bit temporal WEP key, the transmitter MAC address, and the frame sequence number. The function produces a 104-bit per-frame WEP key and 24 bit IV, which are then used to encrypt the frame using the normal WEP encryption procedure.

Although quantitative security analyses of the TKIP mixing are still to be published, cryptographic review suggests that it achieves its design objectives. All documented WEP flaws are fixed by TKIP including key-stream reuse and inherent RC4 vulnerabilities. The use of both the ICV and the MIC to ensure the integrity of 802.11 frames make attacks that alter the encrypted data computationally infeasible. The MIC protection of source and destination addresses together with the data payload ensures that transmissions cannot be spoofed or re-directed to unauthorized destinations. This mitigates the feasibility of both session-hijack and 'Man in the Middle' attacks.

TKIP is a remarkable achievement in that it mitigates WEP flaws without the need for hardware upgrades to 802.11-based infrastructure. However, backward-compatibility requirements have resulted in duplicate security functions and inherent inefficiencies in the TKIP mechanisms. While TKIP is certainly a quick fix to the innate weaknesses of the original 802.11 security specification, it should not be deemed a long-term solution.

### B. CCMP

CCMP (Counter-Mode-CBC-MAC Protocol) has been selected by the IEEE as the long term solution to all known WEP vulnerabilities and forms the basis of the second stage of the IEEE 802.11i Standard. It has not been designed for use on existing hardware and thus existing 802.11-based networks will need to be upgraded if CCMP is to be used. The Advanced Encryption Standard, with 128-bit keys, was selected as the CCMP encryption algorithm.

None of the existing AES modes of operation offered the required balance of features needed to secure the IEEE 802.11 Standard. A new mode called AES-CCM was thus devised. AES-CCM merges two well known Cipher Block Chaining (CBC) security techniques to provide encryption and message integrity in a single solution [3]. The CBC Counter mode (CBC-CTR) algorithm is used to assure confidentiality while the CBC Message Authentication Code (CBC-MAC) algorithm provides integrity protection via a Message Integrity Code (MIC).

AES-CCM uses cryptographically known functions but has the disadvantage of requiring two CBC operations for encryption and message integrity. This is computationally expensive and adds significant overhead to the encryption process. Although CCMP has many similarities to TKIP, freedom from constraints imposed by legacy hardware allows CCMP to possess a more elegant design.

## V. ATTACKING 802-11 BASED NETWORKS

The inherent vulnerabilities and flaws of the WEP encryption process are widely recognized. This section will attempt to document the tools and methods commonly used to attack the security of the WEP encryption algorithm.

The first stage of the WEP cracking process involves obtaining a set of encrypted 802.11 frames. For this to be achieved, the wireless network card must be placed in 'promiscuous' or 'monitor' mode. This mode allows application programs to access the raw bits received by the wireless network adapter. Ethereal [15] is a network monitoring tool that supports the capture and analysis of 802.11 frames using the libpcap library [16]. The Ethereal user interface displays a list of frames received, the frame contents and an interpretation of the information contained in the frame. Complex filtering mechanisms are defined to aid the analysis of captured frames.

There are numerous programs available that are capable of obtaining the WEP encryption key from a set of encrypted frames. The attacks on the WEP encryption algorithm range include both passive and active attacks and each has a number of different implementations.

### A. FMS attack

AirSnort [17] is by far the most popular tool for WEP cracking and also includes wireless sniffing capabilities. AirSnort requires the capture of approximately 1-4 million encrypted frames to successfully crack a WEP key. The program authors estimate that the program requires approximately 1500 weak IVs to recover a 128-bit key and 115 weak IVs per byte for all other key lengths. Use of AirSnort to crack current WEP implementations proved unsuccessful however. AirSnort was unable to crack a 64-bit encryption key even after 2.5 million encrypted frames were captured. Only 301 'Interesting' or weak IVs were discovered, well below the 920 weak IVs that the program creators estimate are required.

This result initially proved very surprising, as this number of captured frames is well above the amount usually needed to crack a 64-bit key. Further investigation revealed that since early 2002, some manufacturers have incorporated an IV filtering mechanism to prevent the use of almost all of the weak IVs normally exploited by AirSnort. It should be noted that this mechanism is only effective when all entities on the network avoid the use of weak IVs.

Unfortunately, the IV filtering mechanism still does not prevent all FMS attacks on the WEP encryption algorithm. In August 2004, a new WEP crack tool called AirCrack [18] was released. AirCrack implements an optimized FMS attack along with 'new advanced WEP attacks' discovered by the hacker 'KoreK'. The program is available for both the Windows and Linux Operating Systems and is able to crack WEP keys using a substantially smaller set of WEP encrypted frames. The program creator estimates that AirCrack requires as little as 200,000 unique IVs compared to over 2 million needed for an effective AirSnort assault. AirCrack is also able to crack 256-bit and 512-bit WEP keys. AirCrack was used to crack both 64-bit and 128-bit encryption keys with relative ease.

## B. Dictionary attack

Dictionary attacks attempt to crack WEP encryption keys that have been generated from a pass-phrase by using knowledge of the key generation algorithm and a list of common passwords. WEP keys generated from each password on the list are used to decrypt a WEP frame with success being confirmed via the frame checksum. The password list used can be extremely comprehensive. A list provided for use with the WEPAttack tool [19] contains 2,854,264 unique pass-phrases in a number of different languages. Use of such lists means that WEP keys generated from most pass-phrases can be cracked relatively easily. Strong pass-phrases are required to prevent such attacks and these should be at least 8 characters in length and must contain a combination of both alphanumeric, upper and lower case and special characters.

## C. Brute Force

The Brute Force form of attack involves decrypting the WEP frame using every possible WEP encryption key and verifying success using the appended checksum. The entire key-space will thus be tested and the WEP key will eventually be recovered. This attack is usually used as a method of last resort due to the time and resources required to carry out the attack. It does however only require a single WEP-encrypted 802.11 frame to be possessed by the attacker. 64-bit WEP encryption (40 effective bits) has a key space of  $2^{40}$  keys. Traversing the key-space at 500,000 keys per second will thus take approximately 25 days. Traversing the 128-bit encryption key-space at the same rate will take about  $1.3 \times 10^{18}$  years! The entire key-space does not need to be tested, however, as the WEP key could be discovered relatively early in the traversal process. Furthermore, the brute force attack could be carried out by a number of different machines, drastically shortening the time needed to crack a 64-bit key to a couple of days. Brute-force attacks of 128-bit WEP keys can be considered infeasible with current technology.

## VI. 802.11 PERFORMANCE ASSESSMENT

The aim of this study is to determine the effect of each of the various 802.11 security mechanisms on the overall performance of the network, as measured at the transport layer of the OSI model. Performance metrics at this layer offer a much more realistic reflection of network efficiency and the amount useful bandwidth available

Measurements were obtained using the NetIQ Chariot software program [14]. The software was installed on each of the endpoints and tests were carried out using modified versions of scripts provided. The Throughput Script (Throughput.scr) file was used to test Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) performance. The script was modified to send a 500,000 byte file between the two endpoints for each timing record, using each of the two transport layer protocols in turn. A hundred timing records were obtained during the transmission period and these were used to obtain throughput and response time measurements.

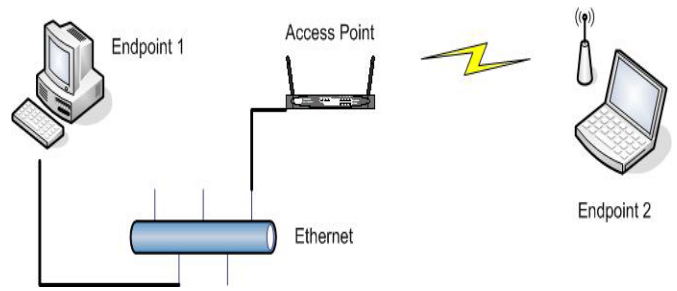


Figure 1: Hardware setup

The network was configured to implement each of the security mechanisms in turn and in isolation. The performance tests were then executed four times to mitigate the effects of spurious interference. The average of each of the performance metrics was then calculated, from which the final results are drawn.

## A. Results

	Avg. TCP (Mbps)		Max. TCP (Mbps)	
1 No Encryption (visible)	23.448	0%	25.000	0%
2 No Encryption (invisible)	23.072	-2%	25.081	0%
3 No Encryption (Access Control)	23.335	0%	25.001	0%
4 WEP-64bit (open authentication)	22.718	-3%	23.988	-4%
5 WEP-64bit (shared key auth)	22.944	-2%	24.480	-2%
6 WEP-128bit (open authentication)	23.040	-2%	24.280	-3%
7 WEP-128bit (shared key auth)	22.470	-4%	24.060	-4%
8 TKIP (PSK)	19.372	-17%	19.951	-20%

Table 1: TCP throughput statistics

Maximum TCP throughput reached 25 Mbps, less than half the specified data transmission rate of the 802.11g Standard. This is to be expected as the rated 54 Mbps refers to the physical transmission of bits and does not take into account frame and packet overhead.

Both closed network authentication (invisible SSID) and MAC access control have little or no effect on network performance.

WEP encryption imposes a 2-4% penalty on TCP throughput, regardless of whether 64-bit or 128-bit keys are used. WEP encryption is implemented in firmware and should thus not add significant overhead to the encryption process.

TKIP encryption using a Pre-Shared Key drastically reduces network performance by as much as 20%. Since the actual encryption stage is exactly the same as the WEP encryption process, it appears that the per-packet keying mechanism and calculation of the Michael MIC reduces TCP throughput by approximately 15%.

## B. Conclusions

Closed network authentication and MAC access control do not impose any penalty on network performance and should thus be implemented wherever feasible. Closed network authentication can be circumvented but presents an effective security barrier to naive users. MAC access control offers a similar level of security but is only useful in small deployments where the overhead of managing the list is not significant. WEP encryption does reduce network

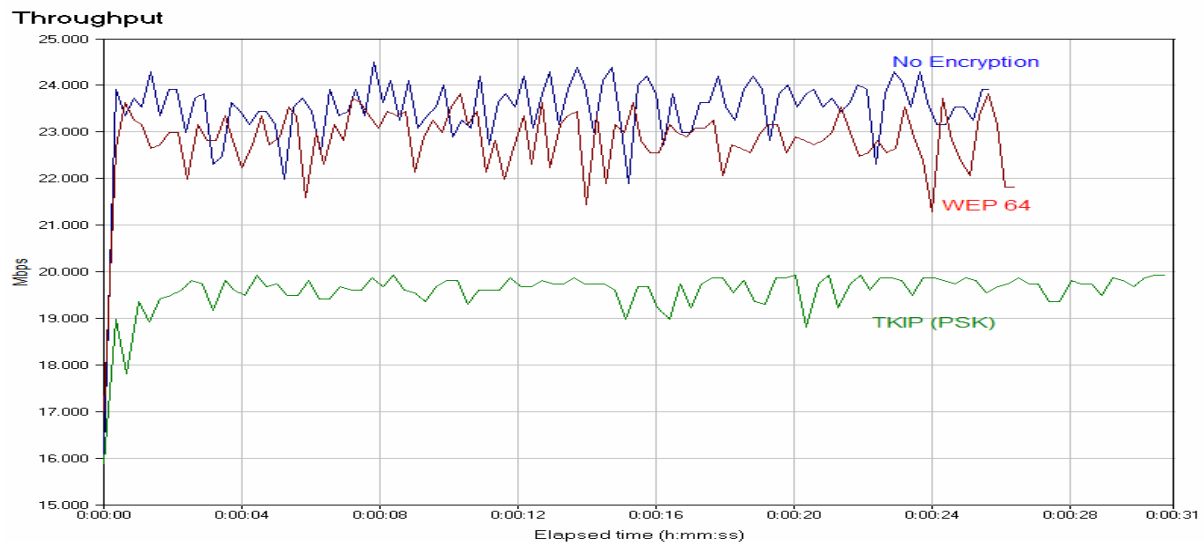


Figure 2: TCP Throughput for 802.11 encryption methods

throughput by a non-negligible amount. However there appears to be no significant effect on performance when 64-bit encryption keys are replaced by 128-bit keys. 128-bit keys offer a slightly higher level of security and should thus be implemented wherever possible. TKIP using a PSK imposes significant overhead and greatly affects network performance. The extra security provided by TKIP encryption should be weighed against the potential performance penalty before implementation.

#### REFERENCES

- [1] W.A. Arbaugh, N. Shankar and Y.J. Wan "Your 802.11 Wireless Network has No Clothes", 2001. [ONLINE] Available: <http://www.cs.umd.edu/~waa/wireless.pdf>
- [2] N. Borisov, I. Goldberg and D. Wagner "Intercepting Mobile Communications: The Insecurity of 802.11", ACM Sigmobile 2001.
- [3] N. Cam-Winget, R. Housley, D. Wagner and J.R. Walker "Security Flaws in 802.11 Data Link Protocols", Communications of the ACM vol. 46, no. 5, 2003.
- [4] N. Ferguson "An improved MIC for 802.11 WEP", 2001. [ONLINE] Available: <http://www.cs.berkeley.edu/~daw/papers/wireless-cacm.pdf>
- [5] S. Fluhrer, I. Mantin and A. Shamir "Weaknesses in the Key Scheduling Algorithm of RC4", Eighth Annual Workshop on Selected Areas in Cryptography 2001.
- [6] M. Gast "InteropNet Labs: TLS and PEAP Comparison - Inner Authentication Methods", 2003. [ONLINE] Available: <http://www.ilabs.interop.net/WLANSec/TLS-PEAP-lv03.pdf>
- [7] D. Halasz "IEEE 802.11i and wireless security", 2004. [ONLINE] Available: <http://www.embedded.com/showArticle.jhtml?articleID=34400002>
- [8] D. Harkins "Attacks against Michael and their Countermeasures", 2003. [ONLINE] Available: <http://www.ieee802.org/11/Documents/DocumentHolder/3-211.zip>
- [9] D. Hulton "Practical Exploitation of RC4 Weaknesses in WEP Environments", 2002. [ONLINE] Available: <http://www.dachb0den.com/projects/bsd-airtools/wepexp.txt>
- [10] InteropNet Labs "What are your EAP Authentication Options?", 2003. [ONLINE] Available: [http://www.ilabs.interop.net/WLANSec/What\\_are\\_EAP\\_types-lv03.pdf](http://www.ilabs.interop.net/WLANSec/What_are_EAP_types-lv03.pdf)

- [11] A. Mishra and W.A. Arbaugh "An Initial Security Analysis of the IEEE 802.1X Standard", 2002. [ONLINE] Available: [http://www.prism.gatech.edu/~gte369k/csc/802\\_1x.pdf](http://www.prism.gatech.edu/~gte369k/csc/802_1x.pdf)
- [12] A. Stubblefield, J. Ioannidis and A.D. Rubin "A Key Recovery Attack on the 802.11b Wired Equivalent Privacy Protocol (WEP)", ACM Transactions on Information and Systems Security vol. 7, no. 2, pp. 319-332, 2004.
- [13] J.R. Walker "Unsafe at any key size: An analysis of the WEP encapsulation", Intel Corporation 2000. [ONLINE] Available: <http://www.dis.org/wl/pdf/unsafe.pdf>
- [14] NetIQ Chariot [ONLINE] <http://www.netiq.com/>
- [15] Ethereal [ONLINE] <http://www.ethereal.com/>
- [16] Libpcap Library [ONLINE] <http://www.tcpdump.org>
- [17] AirSnort [ONLINE] <http://airsnort.shmoo.com/>
- [18] AirCrack [ONLINE] <http://www.cr0.net:8040/code/network/>
- [19] WepAttack [ONLINE] <http://wepattack.sourceforge.net/>

**Sajeet Giga** received his BSc (Eng) degree in Electrical and Computer Engineering with First Class Honours from the University of Cape Town in 2004. In February 2005, he joined the University of Cape Town's Communications Research Group (CRG) in the Department of Electrical Engineering, where he is working towards his MSc (Eng) degree.