Performance of State Based Key Hop (SBKH) Protocol for Security on Wireless Networks

Srinivasan, K.; Mitchell, S.

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

*Performance of State Based Key Hop (SBKH) Protocol for Security on Wireless Networks **

Srinivasan, K., and Mitchell, S.
September 2004

* published at The 60th IEEE Vehicular Technology Conference 2004-Fall (VTC 2004 - Fall). Los Angeles, California, USA. September 26-29, 2004. NRC 47460.

Canada

Performance of State Based Key Hop (SBKH) Protocol for Security on Wireless Networks

Kannan Srinivasan
Institute for Information Technology-Wireless Systems
National Research Council of Canada
Sydney, Canada
Kannan.Srinivasan@nrc-cnrc.gc.ca

Stephen Michell
Institute for Information Technology-Wireless Systems
National Research Council of Canada
Sydney, Canada
Stephen.Michell@nrc-cnrc.gc.ca

Abstract— **State based key hop (SBKH) protocol is a novel, simple encryption scheme that uses RC4 encryption technique in a way that provides robust security with reduced processing cost, compared to 802.11 security protocols: wired equivalent privacy (WEP) and wireless fidelity (Wi-Fi) protected access (WPA). Low processing cost makes SBKH suitable for battery-operated devices such as wireless sensors, and ease of implementation and maintenance make it suitable for small office home office (SOHO) user nodes. In this paper, we present the performance analysis of the state based key hop (SBKH) protocol and compare it with the performance analysis of WEP and WPA. The analysis reveals that SBKH is significantly power efficient in terms of processing compared to WEP, WPA 1.0 and WPA 2.0 for all packet sizes.**

Keywords-WLAN Security, State Based Encryption, Power Efficient Encryption.

I. INTRODUCTION

WEP has been the security standard for 802.11 networks but suffers from significant security issues [FMS 2001, SIR 2001] such as bit-flipping attack, replay attack, weak key attack and forgery attack. These attacks make WEP easy to crack and so a new proposal was needed to provide enhanced security. IEEE 802.11 task group i (802.11 TGi) is working on a new standard (802.11i) [Draft 2003] for security in 802.11 networks. There are two proposals from 802.11 TGi: one for legacy devices and the other for future devices. WPA is a subset of 802.11i and claims to be future compatible with 802.11i. WPA for legacy devices is called WPA 1.0 and WPA for future devices is called WPA 2.0.

WPA 1.0 provides message integrity code (MIC) for data integrity to protect data from forgery and bit-flipping attacks. Michael is the MIC algorithm used in WPA 1.0 that adds an 8 octet field called MIC key to every MAC service data unit (MSDU). Michael is still vulnerable to replay attacks. Temporal key integrity protocol (TKIP) provides extended initialization vector (IV), adding 4 octets of additional overhead to every MAC payload data unit (MPDU), to protect Michael from replay attacks.

WPA 1.0 can work with 802.1x and extensible authentication protocol (EAP) to provide per-node session keys. This mode of operation requires hard-to-maintain RADIUS server in any 802.11 networks. WPA 1.0 has another mode called pre- shared key (PSK) that is being used in many SOHO user nodes in which all the nodes carry static base key. This mode has significant security issues as well [Moskowitz 2003]. It also suffers from attacks from any insider who has the same key.

WPA 1.0, as WEP, still reinitializes RC4 states by running key scheduling algorithm (RC4-KSA) and then carries out the pseudo random number generation algorithm (RC4-PRGA) to generate encryption cipher stream, for every packet. We show that WPA 1.0 due to its complexity, extensive processing cost and high overheads (8 octet MIC key + 4 octet extended IV) is unsuitable for battery-operated devices and SOHO users.

WPA 2.0 is based on advanced encryption standard (AES) encryption scheme. In this paper AES-CCM protocol is assumed for WPA 2.0 which operates under counter (CTR) mode for data encryption and cipher-block chaining (CBC) mode for message integrity check.

Under WPA 2.0, a packet is first split into blocks each of size equal to the AES block size (128 bits) before any encryption starts. This may need padding zeroes to a packet if its size is not an exact multiple of the block size. Following this, AES-CCM generates a block counter and a CBC-IV from a base key, packet sequence counter (48 bits) and MAC header fields. The CBC-IV thus generated is then used along with the base key to carry out AES encryption under CBC mode over plain text payload and portions of MAC header. The end result obtained is truncated to 8 octets and is used as a MIC tag. This MIC tag is then concatenated to the payload and is then encrypted by AES under CTR mode using the block counter as the initial counter. Thus it is obvious that WPA 2.0 runs AES encryption twice for every packet and also adds overhead (8 octet MIC + 4 octet packet number) for every packet, and so can be very expensive computationally. Our new protocol, SBKH [SM 2004], summarized in the following section provides robust security with significantly less processing cost without adding overheads to a packet.

II. SBKH PROTOCOL OVERVIEW

SBKH [SM 2004] protocol uses RC4 in a state based manner that requires minimal processing cost to provide robust security to power constrained nodes and SOHO users.

SBKH works with pairs of communicating nodes sharing common RC4 states for encryption and decryption. SBKH maintains an encryption key pair called as Base Key Pair: one

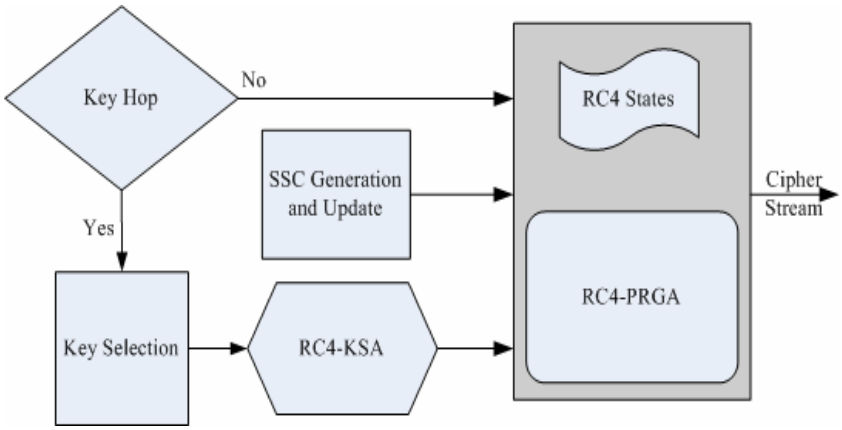

Figure 1. SBKH Protocol Operation

key for the uplink and the other key for the downlink. This Base Key Pair remains unchanged for a duration (called as Time Duration) that is known to both of the communicating nodes. SBKH initializes RC4 states by running RC4-KSA when the Base Key Pair changes and does not reinitialize RC4 states for every packet. Instead RC4-PRGA algorithm alone will be executed for every packet to generate cipher stream that will be used to encrypt that packet. This saves some processing power at every node.

The primary feature of SBKH is that for the encryption and decryption of each subsequent packet destined for the same node, instead of using a new key the RC4 state for the last-used key forms the initial state for RC4-PRGA for the new packet. This state-based encryption forms a strong pseudo random cipher stream known to the communicating pair. The strength of this cipher stream lets us provide more certainty on message transfers and perform inherent message integrity as explained below.

SBKH also discards first few bytes of the cipher stream immediately following the execution of RC4-KSA algorithm. This takes place only when the Base Key changes and is not carried out for every packet. The number of bytes to be discarded is referred to as an Initial Offset (I-Offset) and is shared between the communicating nodes. Discarding first few bytes of the cipher stream eliminates weak key issue [FM 2000, Mantin 2001, Roos 1995] that exists in RC4.

SBKH requires the communicating nodes maintain RC4 states so that encryption and decryption of any packets exchanged between them can be successful. This is referred to as the nodes being State Synchronous. This State Synchronicity may be lost due to failure of nodes to update RC4 states before shutdown or due to acknowledgement spoofing. SBKH includes resynchronization mechanisms to bring back such nodes to State Synchronicity using another offset called Sync-Offset (S-Offset) which is also shared between the communicating nodes.

SBKH uses the 24-bit IV field of the original 802.11 Payload (with WEP) frame [Standard 2001] as a counter called SBKH sequence counter (SSC). This counter is used pair-wise i.e. for every pair of communicating nodes and for each direction of communication (uplink and downlink), a node has to maintain an SSC. This counter is useful in keeping the nodes State Synchronized, and to assist in successful decryption using correct RC4 state in the event of retransmits. Fig. 2 depicts format of an SBKH encrypted packet along with SSC.

SBKH eliminates forgery attacks such as redirection of frames by carrying out pair-wise state based encryption. Redirection of encrypted frames to another node will result in failure of decryption at that node, as the RC4 state for that node would be different from that for the actual destination. For this reason, replay attacks and bit-flipping attacks will also fail. This provides message integrity without the need for an additional message integrity algorithm. A key feature of SBKH is that the RC4 states maintained for different pair of nodes could be different for the same Base Key as the RC4 states depend on the number of bytes of data exchanged between the nodes. This provides some protection from insiders avoiding easy sniffing and active attacks from them, which were possible with WEP and WPA 1.0 under PSK. Thus, SBKH provides robust security without additional overhead and with reduced processing cost. This makes SBKH suitable for battery operated devices and SOHO users.

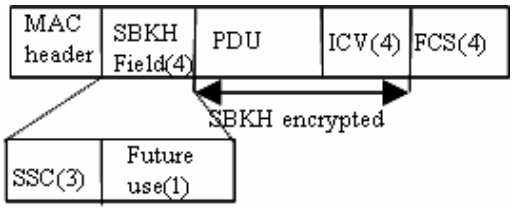

Figure 2. SBKH Encrypted Data Packet

III. PERFORMANCE ANALYSIS

Since power consumed and processing cycles available on standalone devices is a significant issue [Walker 2002], protocols such as SBKH that reduce power are beneficial. Power is used by the processor to execute instructions, by the receiver to receive data and by the transmitter to send data. So

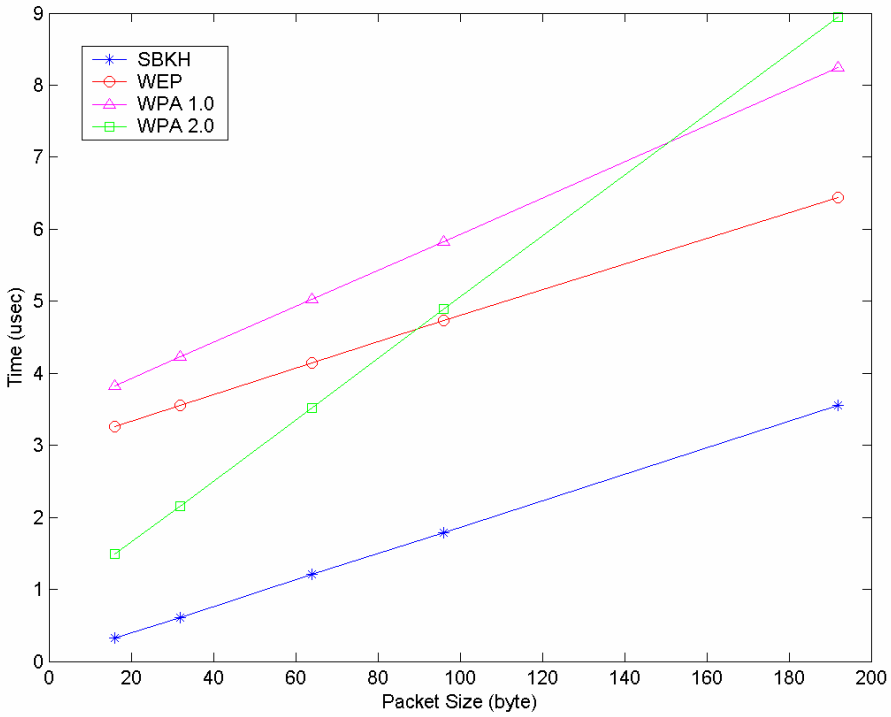

Figure 3. Protocol Processing Cost Comparison for Smaller Packet Sizes

a protocol such as WPA, which adds overhead to encrypted packets and adds processing via TKIP and Michael (WPA 1.0) or AES-CTR and AES-CBC (WPA 2.0), increases power drain, and may also decreases network performance.

SBKH adds no additional overhead (compared to WEP). In addition, SBKH eliminates the RC4-KSA algorithm on all packets except at key change, so should be significantly cheaper, especially for small packets such as is found in sensor networks.

In the following sections we provide the implementation details and discuss the results thus obtained from the performance tests.

IV. IMPLEMENTATION DETAILS

Tests were performed on WEP, WPA 1.0, WPA 2.0 and SBKH. In each case, only the core encryption/decryption protocol was modeled, as the other portions such as authentication, resynchronization and key hopping are executed so infrequently as to make them irrelevant. All tests were written in C language, optimized at level 7, and used OPENSSL 0.9.7 and standard open source CRC algorithms to implement the encryption algorithms. All tests were executed on 1.2 MHz Intel Pentium class processors and 600 MHz Sun SPARC processors. Although some differences will occur when executed on embedded 16-bit processors, we believe that the relative performances will be as described herein.

For each protocol executed, we modeled the execution of state initialization, generation of the CRC for integrity check value (ICV), inclusion of additional octets as specified by the particular algorithm, and final data encryption by RC4-PRGA or AES under CCM mode. For WPA 1.0 and 2.0 we included the IV-setting and creation of the additional encryption. For some of the subtler portions of WPA and AES key generation we used representative code instead of trying to faithfully reproduce the exact algorithms. The algorithms chosen were simpler and more time-efficient hence show WPA 1.0 and 2.0 in better light than the fully implemented algorithms.

The SBKH algorithm was tested exactly as it is specified, including a single RC4-KSA step plus the execution of I-Offset for 5000 Octets for the duration of a single test, and a CRC to generate the ICV for every packet. This represents SBKH executing with an approximate rekey time equivalent to the transmission of one million packets. Smaller rekey time-equivalents or larger offsets would see slightly higher overheads, but we expect the load due to rekeying to remain in the 0-5% range.

All tests were executed for one million iterations multiple times on an unloaded Linux workstation with all windowing and networking disabled. The minimum values were used for each series of runs since these represent optimal executions with respect to cache and any other CPU activity.

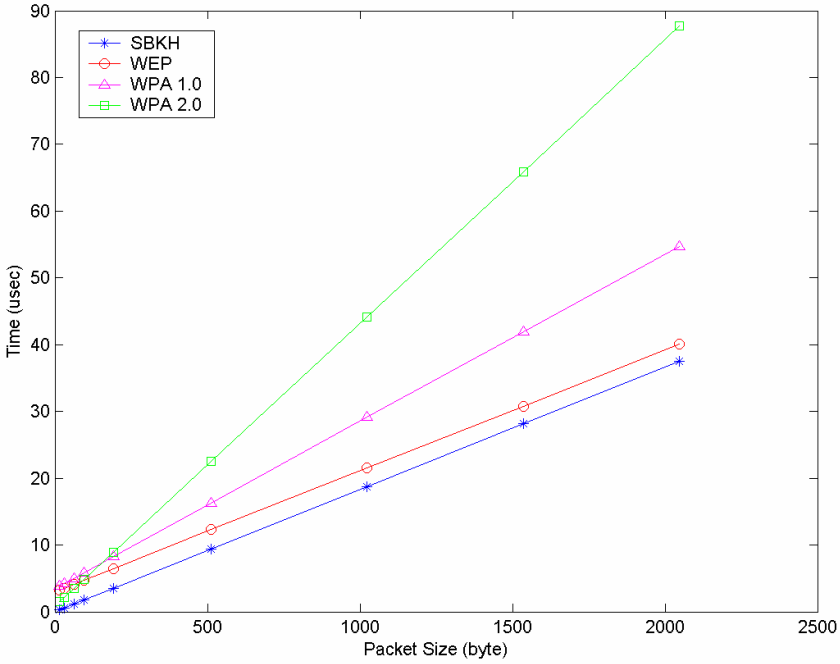

Figure 4. Processing Cost Comparison for Large Packet Sizes

V. IMPLEMENTATION RESULTS

Fig. 3 shows the CPU processing times for SBKH, WEP, WPA 1.0 and WPA 2.0 for packet sizes less than 200 bytes. Fig. 4 shows representative test results for WEP, WPA 1.0, WPA 2.0 and SBKH at representative payloads from 16 octets through 2048 octets. As reported by [PK 2003], we also found that the AES-based WPA 2.0 is significantly more costly than WEP (refered in [PK 2003] as RC4) for packet sizes larger than 90 bytes, but was more efficient below 70 or 80 bytes. Our crossover point for WPA 2.0 is significantly lower than [PK 2003], probably because we implemented the additional message integrity algorithms for WPA 2.0.

The main result from these tests is that SBKH is more efficient than WEP, WPA 1.0 and 2.0 across all packet sizes. For small packets (Fig. 3), which embedded systems expect to use, the results are significant with SBKH being about 75% more efficient than its closest rival. SBKH is similarly more efficient than WEP across the complete payload range, even though it implements a much more robust data encryption algorithm. The reasons for SBKH's superior performance comes from the idea that the shared state lets us avoid some costly additional processing needed by the other protocols to initialize the encryption/decryption states and to provide message integrity of data.

Note that SBKH does not execute RC4-KSA for every packet. RC4-KSA is approximately as costly as executing RC4-PRGA for a 250-300 octet message; hence for small encryptions there is a fixed cost for every packet. This is the reason that WEP and WPA 1.0 are more expensive than WPA 2.0 for packets below 250 octets. SBKH's elimination of this step shows significant savings even when compared to WPA 2.0.

The second reason that SBKH is more efficient is that SBKH does not need the additional message integrity key generation and integrity checking used in WPA 1.0 and 2.0. Since IEEE 802.11 headers are in the clear this integrity checking in WPA 1.0 and 2.0 is carried over portions of IEEE 802.11 header to protect the integrity of data and the header to avoid forgery. SBKH avoids this extra baggage by changing (moving) the encryption stream after each packet, nullifying any attempt at packet capture and retransmission at a later time (any such packets would not successfully decrypt). Similarly, any modifications which change the IEEE 802.11 sender or receiver fields fail decryption because the base key differs for each communicating pair.

In our implementation of WPA 2.0 the payload size is a multiple of the block size and so there was no need to add padded zeroes. This will make the processing cost for WPA 2.0 even more expensive than we find in our implementation for packets with payload size not equal to a multiple of the block size. It should also be noted that additional overhead (12 octets in WPA 1.0 and 8 octets in WPA 2.0) would also cost power in transmission which has not been considered in our implementation. This additional transmission power will not be present in SBKH as SBKH does not have additional overhead (compared to WEP) in packets. Our implementation results hence reflect the best case scenario for WPA 1.0 and 2.0, and still SBKH is significantly cost effective compared to WEP, WPA 1.0 and WPA 2.0.

VI. CONCLUSION AND FUTURE WORK

The processing power comparison carried out in this paper compares CPU time taken to perform encryption/decryption using SBKH, WEP, WPA 1.0 and WPA 2.0. This comparison indicates significant power efficiency using SBKH at all payload sizes compared to existing IEEE 802.11 security proposals namely WEP, WPA 1.0 and WPA 2.0. SBKH carries out state based encryption using RC4 technique and does not require state initialization for every packet which contributes to significant power savings for SBKH. Our implementation did not consider additional power in transmission of additional overhead present in WPA 1.0 and WPA 2.0 which would increase the cost of WPA 1.0 and WPA 2.0 even further. Hence SBKH should be strongly considered as the encryption protocol for power-limited devices that require security such as the wireless sensor nodes.

It should be noted that state based encryption protocols such as SBKH require memory to carry out power efficient encryption and decryption. Our future work will focus on further optimization of SBKH in terms of memory usage. Further investigation of countermeasures, support for broadcast and multicast, and resynchronization of states will be conducted for SBKH. Implementation and testing of SBKH on real hardware will also be carried and will be presented in future papers.

REFERENCES

[Draft 2003] Draft Amendment to Standard for Telecommunications and Information Exchange Between Systems – LAN/MAN Specific Requirements – Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: MAC Security Enhancements, IEEE Std. 802.11i/D7.0, Oct. 2003.

[FM 2000] Fluhrer S. and McGrew D. Statistical Analysis of the Alleged RC4 Keystream Generator, FSE: Fast Software Encryption, FSE2000, Springer-Verlag, 2000.

[FMS 2001] Fluhrer S., Mantin I., Shamir I., Weaknesses in the key scheduling algorithm of RC4, SAC'2001, 2001.

[Mantin 2001] Mantin I., Analysis of the Stream Cipher RC4. Weizmann Institute of Science, Nov. 2001.

[Moskowitz 2003] Moskowitz R., Simple Secrets/Simple Security, ICSA Labs, 2003.

[PK 2003] Prasithsangaree P., Krishnamurthi P., Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs, Global Telecommns. Conf., Globecom '03, Dec. 2003.

[Roos 1995] Roos A., A Class of Weak Keys in the RC4 Stream Cipher. sci.crypt posting, Sept. 1995.

[SM 2004] Srinivasan K., Michell S., State Based Key Hop Protocol, Proc. of The Sixteenth International Conf. on Wireless Commns., Wireless 2004, Jul. 2004.

[Standard 2001] IEEE Standard for Local and metropolitan area networks – Port-Based Network Access Control, Oct. 2001.

[SIR 2001] Stubblefield A., Ioannidis J., and Rubin A. D., Using the Fluhrer, Mantin, and Shamir Attack to Break WEP, AT&T Labs Technical Report TD-4ZCPZZ, Aug. 2001.

[SPIN 2003] The SPIN Model Checker: Primer and Reference Manual, Addison Wesley, 2003.

[Walker 2002] Walker J., 802.11 Security Series – Part II: The Temporal Key Integrity Protocol (TKIP), Intel Corporation, 2002.