Wireless LAN Security Solution Motives and Rationale

NIST WLAN Security Workshop

4-5 December 2002

Vigil Security LLC

Russ Housley

housley@vigilsec.com

Outline

- Packet Security
  - TKIP
  - CCMP
- Key Management

Packet Security

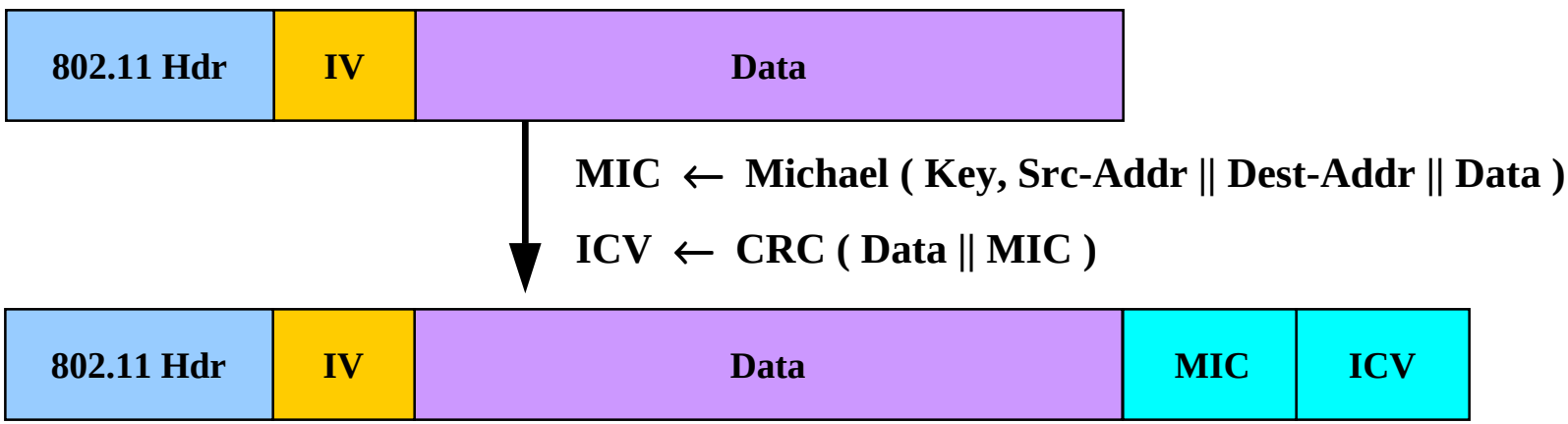
- Solutions must address *all* of the problems, otherwise new attack tools will be developed to exploit the remaining holes
  - IV Collisions
  - Weak Keys
  - Message Forgery
  - Replay

TKIP Mechanism Review

- Message Integrity Code (MIC) called Michael
- New per-packet key derivation function, with a large IV
- IV Sequencing, with a large IV

Michael

- Compute MIC using Michael (a new algorithm)
  - Designed for deployed hardware by Niels Ferguson
    - 3-4 cycles/byte on ARM7
    - 5-6 cycles/byte on i486
  - Provides about 30 bits of security – best possible in time budget
- The most critical step – it takes away active attacks



Michael Performance

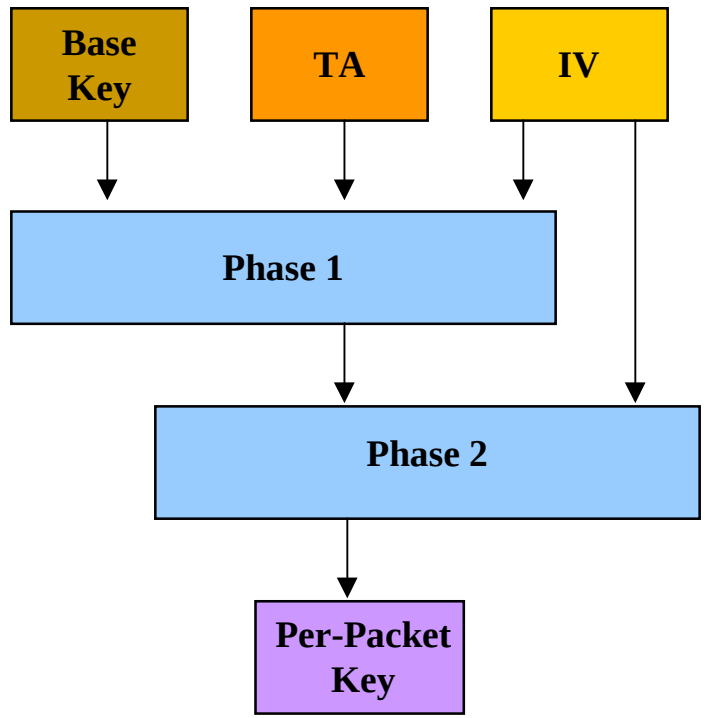
- Allow implementation on host or NIC
- Host implementation
  - Acceptable performance
  - Impacts host-to-NIC interface
- NIC implementation
  - Acceptable performance
    - Already running at 90% CPU utilization
  - No impact on host-to-NIC interface

Michael Security

- Tradeoff between performance and security
- About 30 bits of security – would like more
  - Use FCS and ICV to eliminate damaged packets
  - Use countermeasures to improve security
- 64 bits of overhead – would like less
- Increasing security or decreasing size of the MIC would require more cycles
  - Vendors are already uncomfortable with the number of cycles needed to compute the MIC
- Cover source and destination addresses in addition to the packet payload

Per-packet Key Derivation Function

- Compute Per-packet Key from:
  - 128-bit Base Key
  - 48-bit Transmitter Address (TA)
  - 48-bit IV
- Structure permits:
  - Efficient on deployed hardware
  - Supports caching and precomputation
- Avoids the weak keys exploited by AirSnort and other hacker tools



KDF Performance

- Target NIC implementation
- Permit precomputation
  - Sender will not waste precomputation
  - Receiver might waste precomputation
- Cache Phase 1 output, use it for 2^{16} packets

KDF Security

- Bijective Function to ensure that distinct inputs cannot generate the same RC4 per-packet key
- Structure of second octet avoids FMS weak keys
- 48-bit IV to ensures key space will not be exhausted
- Transmitter address
  - Ensures different stations using same key will generate different RC4 per-packet keys
  - Precomputation attack must be targeted at one device

IV Sequencing

- IV management rules:
  - Reinitialize IV to 0 when the base key is established
  - IV is a strictly increasing counter
  - Data traffic halts if IV value reaches maximum value
  - Receiver discards any packets associated with the same key when the IV value is less than a previously received packet

IV ← IV + 1

IV Sequencing Discussion

- Detect Replay
- 48-bit IV ensures that the sequence space will not be exhausted
  - At IEEE 802.11a rates, it would take about 1090 years to exhaust the IV space
- Minimal memory requirement for sender and receiver
- Use IEEE 802.1X EAPOL Key message to establish a new key at start of every association

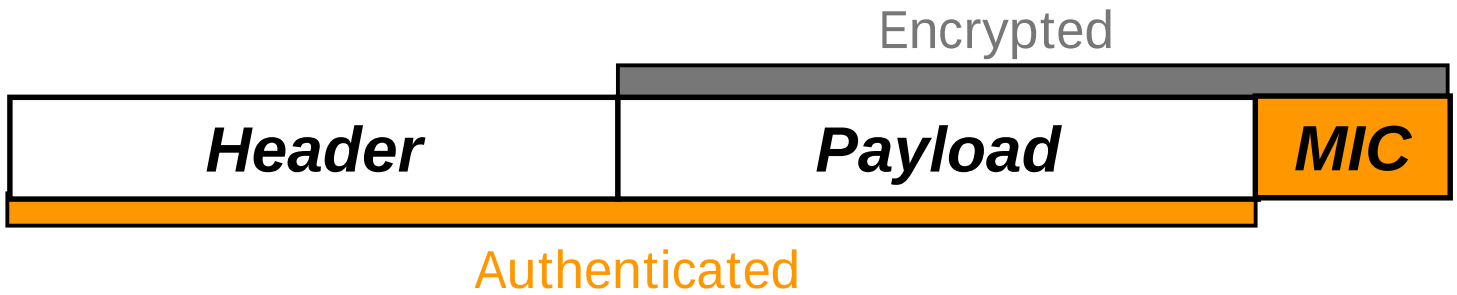
TKIP Anomalies

- 64-bit Michael MIC covers MSDU
- 128-bit encryption covers PDU
- Receiver increments IV before MIC check
  - Allows MIC to be implemented in host and IV processing to be implemented in the NIC
  - MSDU can be fragmented into multiple PDUs, each with a distinct IV value

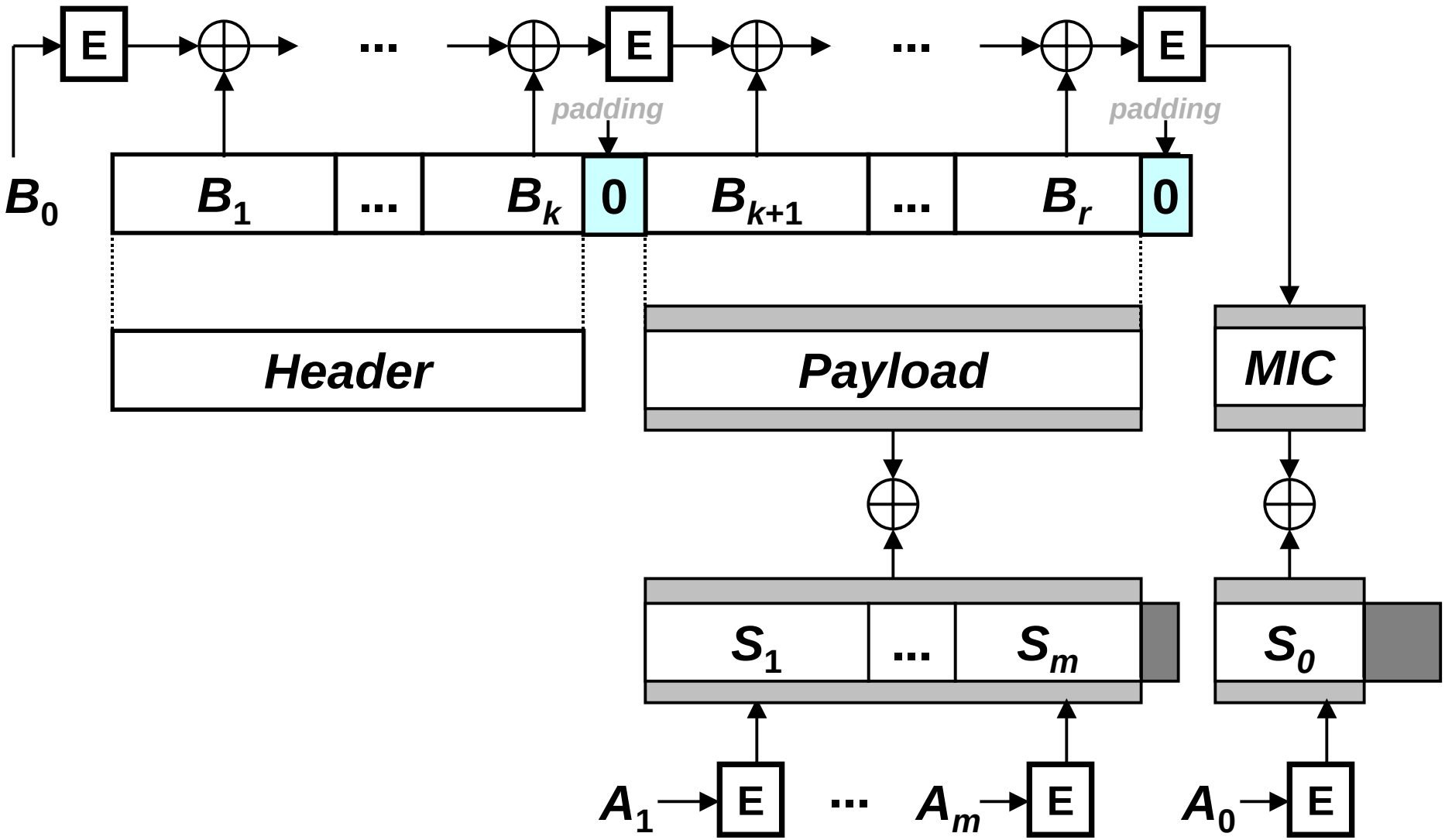
CCMP Mechanism Review

- 48-bit IV used for replay detection
  - First four bits of IV indicate QoS traffic class
  - Remaining 44 bits used as counter
  - Decryption/integrity check fail if traffic class bits are altered
  - Sender uses single counter space, but receiver needs one for each traffic class
- AES with CCM authenticated encryption
  - Header authentication
  - Payload authentication and confidentiality

CCM Mode Overview



- Use CBC-MAC to compute a MIC on the plaintext header, length of the plaintext header, and the payload
- Use CTR mode to encrypt the payload
  - Counter values 1, 2, 3, …
- Use CTR mode to encrypt the MIC
  - Counter value 0


E
E
E
padding
padding
B_0
B_1
...
B_k
0
B_{k+1}
...
B_r
0
Header
Payload
MIC
S_1
...
S_m
S_0
A_1
E
...
A_m
E
A_0
E

CCM Parameters

- Using IEEE 802.11i parameter choices:
  - 16 bits for packet length / counter
  - 64 bits for MIC
- Maximum packets size: 2^{16} octets
  - CTR Mode: 16 * (2^{16} –1) octets [1,048,560 octets]
  - CBC-MAC: (2^{16} –1) octets [65,535 octets]
- Maximum packets per key: 2^{44}
  - 281,474,976,710,656/16 packets per key
  - At 10K packets/sec, rekey in 100 years

CCM Performance

- CCM uses only AES encryption operations, *no* decryption operations
  - Allows smaller implementation size
- Most AES implementations will use hardware, which allows pipelining of Counter mode and CBC-MAC
- In software, CCM requires two passes
  - Not considered onerous
- CCM covers an *arbitrary* amount of cleartext header in addition to the payload
  - Future proof against other 802.11 activities

CCM Security

- Jakob Jonsson did security proof of CCM
  - Published last August at SAC ’02
- Proof shows that CCM provides a level of confidentiality and authenticity comparable to other proposed authenticated encryption modes, such as OCB mode

CCM Patent Status

- Authors explicitly released any intellectual property rights to CCM to the public domain
  - Authors are not aware of any patent or patent application anywhere in the world that covers CCM
  - Authors believe that CCM is a simple combination of well-established techniques; it is obvious to a person of ordinary skill in the arts
- Alternative authenticated encryption modes are encumbered

Key Management Review

- TKIP and CCMP use the same key management
  - Pairwise keys for point-to-point communication
  - Group keys for multicast and broadcast
  - Key wrapping algorithm is coupled to use
    - TKIP key wrapping uses RC4
    - CCMP key wrapping uses AES (NIST Key Wrap)
- No new protocols are being developed
  - EAP
  - IEEE 802.1X

Implementation Reality

- Key management will run as an application
- Application will download keys to network interface card (NIC)
  - Microsoft has defined driver commands to add and delete keys
  - Microsoft has an implementation which uses a RADIUS server for authentication
    - Supports TKIP and CCMP
  - NIC vendors are doing interoperability testing

Political Reality

- IEEE 802.11 is chartered to work in the PHY and MAC
- Key management protocol out of scope
- Must rely on other groups for protocol
- Once key is in place, IEEE 802.11can specify the way that it is used (e.g., PRF)

IEEE 802.1X

- Does not dictate any authentication method
  - Possible role for WiFi Alliance to ensure interoperability
- Arbaugh Attack: IEEE 802.1 is adding state machine synchronization point
- EAPOL packets are *always* authorized
  - Ethertype 88-8E is never blocked
  - Encrypted once PTK is in place

Tunneled Authentication

- Man-in-the-Middle attack
  - Explained in earlier briefing by Jesse Walker
- Only the server is authenticated within the tunnel, which may improve legacy authentication mechanisms that do not generate keys
- Tunneling a well-designed authentication protocol introduces a new vulnerability
- Authentication tunneling is *not* a miracle cure

Pseudo Random Function

- PRF with outputs of 128, 192, 256, 384, and 512

```
PRF(K, A, B, Len)
    for i ← 0 to (Len+159)/160 do
        R ← R || HMAC-SHA-1(K, A || Zero || B || i)
    return substr(R, 0, Len)
```

- The use of HMAC-SHA-1 is based on IPsec
  - Inefficient
  - Security constrained by size of SHA-1 output (160 bits)

Further Discussion