



# Wi-Fi security

The good, the bad and... the ugly...

Cédric Blancher

cedric.blancher@eads.net  
Computer Security Research Lab  
EADS Innovation Works

sid@rstack.org  
Rstack Team  
<http://sid.rstack.org/>

SecurityByte 2009 - Delhi - November 17th-18th 2009  
<http://securitybyte.org/>



## Agenda

- 1 The good : WPA/WPA2
- 2 The bad : vulnerabilities in WPA/WPA2
  - PSK cracking analysis
  - TKIP issues
- 3 The ugly : things we see in the wild



## Who am I?

- Senior Research Engineer at EADS Innovation Works
- Head of Computer Security research lab for 3 years
- Leading Security Evaluation activities

## Wi-Fi Security ?

- Working on Wi-Fi security for 6 years
- Tons of Wi-Fi pentests in various environments
- Numerous talks and trainings
- Traffic injection attacks PoC (Wifitap)



## Introduction

### Wi-Fi security...

- WEP is crippled and broken
- WPA came up to replace it
- Now, we have WPA2



## Introduction

Wi-Fi security...

- WEP is crippled and broken
- WPA came up to replace it
- Now, we have WPA2

And...

- Nothing is perfect for sure
- But we still can get a very decent level of security

Do we get it ?



## Agenda

- 1 The good : WPA/WPA2
- 2 The bad : vulnerabilities in WPA/WPA2
  - PSK cracking analysis
  - TKIP issues
- 3 The ugly : things we see in the wild



## Wi-Fi security

Since 2004, 802.11i/WPA2 is available and mandatory for Wi-Fi certification

- PSK authentication for home users
- 802.1x/EAP based authentication for enterprises
- TKIP based encryption for backward compatibility
- AES encryption as a new standard

## Wi-Fi security

Since 2004, 802.11i/WPA2 is available and mandatory for Wi-Fi certification

- PSK authentication for home users
- 802.1x/EAP based authentication for enterprises
- TKIP based encryption for backward compatibility
- AES encryption as a new standard

### Security ?

All this offers a flexible, decent to excellent level of security



## Better than wire ?

Sometimes, you can get better security than wire

- 802.1x initially defined for wire
- But no encryption is done



## Better than wire ?

Sometimes, you can get better security than wire

- 802.1x initially defined for wire
- But no encryption is done

### Caveats

Easy "Hub + MAC spoofing" trick allows for wired 802.1x bypass



## More to come...

Management traffic still unprotected

- Unauthenticated probe traffic
- Unauthenticated deauth/deassoc
- Large rand of possible attacks



## More to come...

Management traffic still unprotected

- Unauthenticated probe traffic
- Unauthenticated deauth/deassoc
- Large rand of possible attacks

### Solution ?

Upcoming 802.11w should define mechanisms for authenticating management traffic (and deprecating TKIP ?)



## Question...

How come we don't feel that comfortable with Wi-Fi?

- Weak modes (open network)
- Configuration issues
- Wi-Fi vulnerabilities



## Question...

How come we don't feel that comfortable with Wi-Fi?

- Weak modes (open network)
- Configuration issues
- Wi-Fi vulnerabilities

More to come :)



## Agenda

- 1 The good : WPA/WPA2
- 2 The bad : vulnerabilities in WPA/WPA2
  - PSK cracking analysis
  - TKIP issues
- 3 The ugly : things we see in the wild



## A bit of history

So I am at BA-Con talking with Simon on Wi-Fi authentication security

- Demonstrating feasibility of PSK cracking
- Showing some research opportunities in EAP



## A bit of history

So I am at BA-Con talking with Simon on Wi-Fi authentication security

- Demonstrating feasibility of PSK cracking
- Showing some research opportunities in EAP

### When...

- Dragos gets a submission claiming WPA gone
- A week later, PSK bruteforcing breakthrough comes out



## A bit of history

So I am at BA-Con talking with Simon on Wi-Fi authentication security

- Demonstrating feasibility of PSK cracking
- Showing some research opportunities in EAP

### When...

- Dragos gets a submission claiming WPA gone
- A week later, PSK bruteforcing breakthrough comes out
- OMFG, looks like someone just ruined our talk !...



## Food for thought

Cool, some interesting piece of research to look at

- PSK bruteforcing soon being comparable at WEP cracking?
- Breaking TKIP allows for ruining a dramatic load of networks!



## Food for thought

Cool, some interesting piece of research to look at

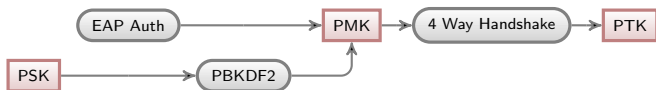
- PSK bruteforcing soon being comparable at WEP cracking?
- Breaking TKIP allows for ruining a dramatic load of networks!

### Outcome

- The good : PSK cracking has not really been improved
- The bad : TKIP vulns, while having limited impact, work
- The ugly : press coverage :)

## Authentication modes

- Preshared secret (PSK)
- EAP



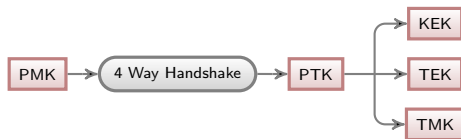
### Key hierarchy

- Authentication leads to Master Key (MK)
- Pairwise Master Key (PMK) derived from MK

## One key to rule them all...

From MK come all further keys

- Pairwise Master Key
- Key exchange keys
- Encryption keys
- Authentication keys if applicable

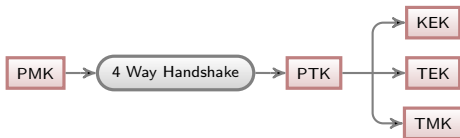




## One key to rule them all...

From MK come all further keys

- Pairwise Master Key
- Key exchange keys
- Encryption keys
- Authentication keys if applicable



Conclusion

Owning the Master Key == Owning everything else

## The Preshared Key option

- MK is your PSK
- PMK is derived from MK



## The Preshared Key option

- MK is your PSK
- PMK is derived from MK



Why cracking PSK is interesting

Owning the PSK == Owning everything



## From PSK to PMK

### PSK is your master key (MK)

- It is your secret key, password or passphrase, i.e. PSK
- 8 to 63 printable ASCII characters (between code 32 and 126)



## From PSK to PMK

### PSK is your master key (MK)

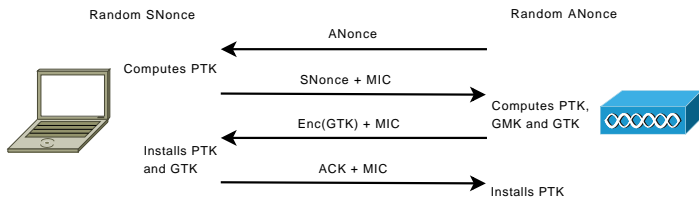
- It is your secret key, password or passphrase, i.e. PSK
- 8 to 63 printable ASCII characters (between code 32 and 126)

### The pairwise master key (PMK)

- $PMK = PBKDF2(SSID, PSK)$
- PBKDF2 function is designed to be time consuming

## Four-Way Handshake

Four-way handshake provides authentication step for PSK mode and PTK computation



MICs are computed against PTK

## Attacking PSK

### Retrieving the relevant data

- It must be captured during the handshake
- It is possible to force this handshake
- Only works for a single SSID

## Attacking PSK

### Retrieving the relevant data

- It must be captured during the handshake
- It is possible to force this handshake
- Only works for a single SSID

### Testing a preshared key

## Attacking PSK

### Retrieving the relevant data

- It must be captured during the handshake
- It is possible to force this handshake
- Only works for a single SSID

### Testing a preshared key

- 1 For every potential PSK, compute the corresponding PMK (PBKDF2 using SSID and PSK)



## Attacking PSK

### Retrieving the relevant data

- It must be captured during the handshake
- It is possible to force this handshake
- Only works for a single SSID

### Testing a preshared key

- 1 For every potential PSK, compute the corresponding PMK (PBKDF2 using SSID and PSK)
- 2 Compute the PTK (4xHMAC-SHA1 using PMK and nonces)

## Attacking PSK

### Retrieving the relevant data

- It must be captured during the handshake
- It is possible to force this handshake
- Only works for a single SSID

### Testing a preshared key

- 1 For every potential PSK, compute the corresponding PMK (PBKDF2 using SSID and PSK)
- 2 Compute the PTK (4xHMAC-SHA1 using PMK and nonces)
- 3 Finally, get the MIC (1xHMAC-SHA1 or MD5) and compare it with the captured handshake



## The PBKDF2 function

### Algorithm of PBKDF2

```
x1 = HMAC_SHA1(MK, SSID + '\1');  
x2 = HMAC_SHA1(MK, SSID + '\2');  
for(i=1;i<4096;i++) {  
    x1 = HMAC_SHA1(MK, x1);  
    x2 = HMAC_SHA1(MK, x2);  
}  
return x1 + x2;
```



## Attack cost

PBKDF2 requires 8192 calls to HMAC-SHA1, plus at least 8 to get MIC

- HMAC-SHA1 needs 2 SHA1
- SHA1 body needs 961 operations
- Attack needs about 15,7M operation per PSK



## Attack cost

PBKDF2 requires 8192 calls to HMAC-SHA1, plus at least 8 to get MIC

- HMAC-SHA1 needs 2 SHA1
- SHA1 body needs 961 operations
- Attack needs about 15,7M operation per PSK

### Comparison with MD5

Testing one MD5 for cracking requires 317 operations...



## Comparing implementations

### PBKDF2 function theoretical speed

- Perfect SSE2, 3Ghz single x86 core : 500 checks/s
- Perfect native CELL : 2,840 checks/s
- Perfect Linux CELL : 2,440 checks/s



## Comparing implementations

### PBKDF2 function theoretical speed

- Perfect SSE2, 3Ghz single x86 core : 500 checks/s
- Perfect native CELL : 2,840 checks/s
- Perfect Linux CELL : 2,440 checks/s

Real world implementations...



## Comparing implementations

### PBKDF2 function theoretical speed

- Perfect SSE2, 3Ghz single x86 core : 500 checks/s
- Perfect native CELL : 2,840 checks/s
- Perfect Linux CELL : 2,440 checks/s

### Real world implementations...

- Aircrack on Xeon E5405 (4x2Ghz) : 650 checks/s



## Comparing implementations

### PBKDF2 function theoretical speed

- Perfect SSE2, 3Ghz single x86 core : 500 checks/s
- Perfect native CELL : 2,840 checks/s
- Perfect Linux CELL : 2,440 checks/s

### Real world implementations...

- Aircrack on Xeon E5405 (4x2Ghz) : 650 checks/s
- Pico Computing FX60 FPGA : 1,000 checks/s



## Comparing implementations

### PBKDF2 function theoretical speed

- Perfect SSE2, 3Ghz single x86 core : 500 checks/s
- Perfect native CELL : 2,840 checks/s
- Perfect Linux CELL : 2,440 checks/s

### Real world implementations...

- Aircrack on Xeon E5405 (4x2Ghz) : 650 checks/s
- Pico Computing FX60 FPGA : 1,000 checks/s
- CELL : 2,300 checks/s (and you can play MGS4)



## Comparing implementations

### PBKDF2 function theoretical speed

- Perfect SSE2, 3Ghz single x86 core : 500 checks/s
- Perfect native CELL : 2,840 checks/s
- Perfect Linux CELL : 2,440 checks/s

### Real world implementations...

- Aircrack on Xeon E5405 (4x2Ghz) : 650 checks/s
- Pico Computing FX60 FPGA : 1,000 checks/s
- CELL : 2,300 checks/s (and you can play MGS4)
- CUDA on 295GTX : 20,000 check/s



## Bruteforcing PSK ?

Is it just possible ?

- PSK space is huge
- PBKDF2 is slow
- 100x factor just means cracking one more character...



## Bruteforcing PSK ?

Is it just possible ?

- PSK space is huge
- PBKDF2 is slow
- 100x factor just means cracking one more character...

Cracking time expectations...

At 100,000 check/s, expect :



## Bruteforcing PSK ?

Is it just possible ?

- PSK space is huge
- PBKDF2 is slow
- 100x factor just means cracking one more character...

### Cracking time expectations...

At 100,000 check/s, expect :

- 34 years to crack alphanumeric 8 chars



## Bruteforcing PSK ?

Is it just possible ?

- PSK space is huge
- PBKDF2 is slow
- 100x factor just means cracking one more character...

### Cracking time expectations...

At 100,000 check/s, expect :

- 34 years to crack alphanumeric 8 chars
- 1051 years to crack full space 8 chars



## Bruteforcing PSK ?

Is it just possible ?

- PSK space is huge
- PBKDF2 is slow
- 100x factor just means cracking one more character...

### Cracking time expectations...

At 100,000 check/s, expect :

- 34 years to crack alphanumeric 8 chars
- 1051 years to crack full space 8 chars
- 2146 years to crack alphanumeric 9 chars



## Bruteforcing PSK ?

Is it just possible ?

- PSK space is huge
- PBKDF2 is slow
- 100x factor just means cracking one more character...

### Cracking time expectations...

At 100,000 check/s, expect :

- 34 years to crack alphanumeric 8 chars
- 1051 years to crack full space 8 chars
- 2146 years to crack alphanumeric 9 chars

Comparison : MD5 cracking can reach 900M checks/s on 295GTX...



## Other cracking methods

Some press/blog/buzz on PSK cracking



## Other cracking methods

Some press/blog/buzz on PSK cracking

### WPA-PSK "rainbow tables"

- Really PMK lookup tables
- Precomputation of 1M passwords for 1000 SSIDs



## Other cracking methods

Some press/blog/buzz on PSK cracking

### WPA-PSK "rainbow tables"

- Really PMK lookup tables
- Precomputation of 1M passwords for 1000 SSIDs

### Lockheed CELL implementation

- *"Lockheed Breaks WPA-Encrypted Wireless Network With 8 Clustered Sony PlayStations"*
- Why bother, done already :/
- Unknown performance



## The best ?

Elcomsoft announces distributed CUDA based cracking

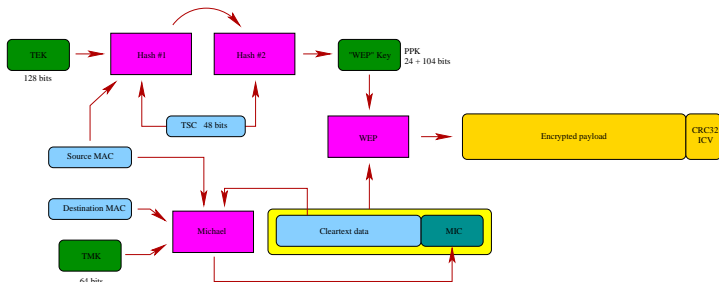
### Elcomsoft PSK bruteforcer

- *"It is now possible to crack WPA and WPA2 protection on Wi-Fi networks up to 100 times quicker"*
- PR bullshit
- Not better than state of the art implementations



## TKIP detailed (or so)...

TKIP is a key scheduling algorithm working over WEP  
Integrity is ensured using Michael MIC



TSC is a frame sequence counter against replay



Let's be more cryptic...

TKIP provides per frame key (PPK) and MIC



## Let's be more cryptic...

TKIP provides per frame key (PPK) and MIC

### PPK computation

$$TTAK_n = MP1(TA, TEK, TSC_n)$$

$$PPK_n = MP2(TEK, TSC_n, TTAK_n)$$



## Let's be more cryptic...

TKIP provides per frame key (PPK) and MIC

### PPK computation

$$TTAK_n = MP1(TA, TEK, TSC_n)$$

$$PPK_n = MP2(TEK, TSC_n, TTAK_n)$$

### MIC computation

$$MIC_n = Michael(DA, SA, MSDU_n, MK)$$

## Let's be more cryptic...

TKIP provides per frame key (PPK) and MIC

### PPK computation

$$TTAK_n = MP1(TA, TEK, TSC_n)$$

$$PPK_n = MP2(TEK, TSC_n, TTAK_n)$$

### MIC computation

$$MIC_n = Michael(DA, SA, MSDU_n, MK)$$

### Frame generation

$$MPDU = WEP(PPK_n, P_n)$$

$$= WEP((IV_n, K_n), P_n)$$

$$= RC4(IV_n \parallel K_n) \oplus (P_n \parallel CRC32(P_n))$$



## Chopchop attack for TKIP

Beck and Tews work starts with chopchop attack

- When it happens, counter-measures are triggered
- When detected by STA, MIC-Failure message sent
- MIC-Failure means "chopchop successful"

## Chopchop attack for TKIP

Beck and Tews work starts with chopchop attack

- When it happens, counter-measures are triggered
- When detected by STA, MIC-Failure message sent
- MIC-Failure means "chopchop successful"

### Limitations

- Only works against AP to STA traffic
- Only 1 MIC-Failure allowed every 60s

## Chopchop attack for TKIP

Beck and Tews work starts with chopchop attack

- When it happens, counter-measures are triggered
- When detected by STA, MIC-Failure message sent
- MIC-Failure means "chopchop successful"

### Limitations

- Only works against AP to STA traffic
- Only 1 MIC-Failure allowed every 60s

Attack should fail anyway because of TSC reuse...



## MIC key retrieval

If he can manage TSC, attacker can decrypt a frame,  
1 byte every 60s

- Fully decrypted frame gives payload+MIC
- Michael is weak so you can crack MIC
- Known cleartext gives you keystream



## MIC key retrieval

If he can manage TSC, attacker can decrypt a frame,  
1 byte every 60s

- Fully decrypted frame gives payload+MIC
- Michael is weak so you can crack MIC
- Known cleartext gives you keystream

Yes but...

Keystream is tight to current TSC that can't be reused...



## Bypassing TSC limitation

802.11e defines QoS standard referred as WME

- Traffic can be characterized in terms of QoS
- 8 queues are available to enforce QoS
- Each queue has its own TSC



## Bypassing TSC limitation

802.11e defines QoS standard referred as WME

- Traffic can be characterized in terms of QoS
- 8 queues are available to enforce QoS
- Each queue has its own TSC

### TSC reuse...

We can replay frames on a different queue

- To perform chopchop attack
- To inject arbitrary data



## Getting rid of 802.11e using MitM

Japanese researchers propose using MitM to bypass TSC

- MitM allows for attacker to block communication
- AP frames are not more transmitted to target
- TSC gets frozen



## Getting rid of 802.11e using MitM

Japanese researchers propose using MitM to bypass TSC

- MitM allows for attacker to block communication
- AP frames are not more transmitted to target
- TSC gets frozen

### Issues

- Requires non trivial (but possible) physical MitM
- Post-attack AP-STA resync to be done



## Norway comes into play

Norwegian researchers improve the attack

- Looking for longer messages
- DHCP traffic is a good candidate
- Up to 585 bytes



## Norway comes into play

Norwegian researchers improve the attack

- Looking for longer messages
- DHCP traffic is a good candidate
- Up to 585 bytes

### Impact

- Longer keystream retrieved
- Same exploitation conditions
- Same impact



## Attacks Summary

TKIP attacks works

- Decrypt predictable frames sent by AP to STA
- At a rate of 1 byte every minute
- Allows for sending up to 7 arbitrary frames to STA

## Attacks Summary

### TKIP attacks works

- Decrypt predictable frames sent by AP to STA
- At a rate of 1 byte every minute
- Allows for sending up to 7 arbitrary frames to STA

### Limitations

- Attack is slow
- Limited to one STA at a time



## Teasing vs. reality

WPA certainly not gone

- Really cool piece of work
- Should ring a bell : move to AES-CCMP asap



## Teasing vs. reality

WPA certainly not gone

- Really cool piece of work
- Should ring a bell : move to AES-CCMP asap

But...

- Affects TKIP only and WPA supports AES
- WPA2 supports TKIP, thus affected too
- Definitely no Man in the Middle



## Agenda

- 1 The good : WPA/WPA2
- 2 The bad : vulnerabilities in WPA/WPA2
  - PSK cracking analysis
  - TKIP issues
- 3 The ugly : things we see in the wild



## Random thoughts

World is full of Wi-Fi security experts unless...

- it comes to differentiate WPA and WPA2
- it comes to make the difference between authentication stage and encryption



## Random thoughts

World is full of Wi-Fi security experts unless...

- it comes to differentiate WPA and WPA2
- it comes to make the difference between authentication stage and encryption
- Stuff like figuring out that  $WPA \neq PSK + TKIP...$



## Random thoughts

World is full of Wi-Fi security experts unless...

- it comes to differentiate WPA and WPA2
- it comes to make the difference between authentication stage and encryption
- Stuff like figuring out that  $WPA \neq PSK + TKIP...$
- Or that now  $WPA \simeq WPA2...$

### All about claims!

- Press will carry your word without question
- You can even get to a conference with false claims...



## Open networks

Still preferred guest access method

- Unauthenticated, unencrypted access
- Captive portal provides access control (webapp)

## Open networks

Still preferred guest access method

- Unauthenticated, unencrypted access
- Captive portal provides access control (webapp)

### Beware...

- Clients are naked
- Clients can be spoofed
- Infrastructure can be abused and/or compromised !



## WPA/WPA2-EAP deployments

802.1x/EAP should provide top of the notch security for Wi-Fi, unless...

- Credentials are compromised (evil maid)
- Credentials are badly checked (x509 verification)
- Infrastructure is badly designed



## WPA/WPA2-EAP deployments

802.1x/EAP should provide top of the notch security for Wi-Fi, unless...

- Credentials are compromised (evil maid)
- Credentials are badly checked (x509 verification)
- Infrastructure is badly designed

### Issues...

- Architectural flaws + config mistakes
- No RADIUS certificate check
- Lax certificate checks
- Vulnerabilities at various levels

So many things to double check...



# The end...

Special thank to Simon Marechal for his work on password cracking!

Thank you all for your attention

Questions?