Security and Cryptography 2022

Mirosław Kutyłowski

XIII. WIFI

standards:

- evolution
- little interaction with academic community
- underspecified,
- sometimes not literally implemented, lack of documentation
- sometimes formal security proofs – like for WPA, but nevertheless ... attacks

Krack against WPA2

- attack based on crypto assumption: “no IV used twice”
- works despite “provable security”, but the proofs do not model all scenarios
- effects depend on particular implementation. Most cases:
  - decryption due to reuse of the same string in stream cipher
  - or just making mess by replay attack (e.g. against NTP- network time protocol)

4-way handshake

- ”supplicant”= user, “authenticator”=Access Point
- PMK Pairwise Master Key is preshared
- PTK (Pairwise Transient Key) derived as a session key
- PTK=$f(\mathrm{PMK}, \mathrm{ANonce}, \mathrm{SNonce})$, PTK splitted into TK (Temporal Key), KCK (Key Confirmation Key), KEK (Key Encryption Key)
- for WPA2 also GPK (Group Temporal Key) transported to the supplicant (used by AP for broadcast)

- frames: EAPOL consisting of
  - **header** - determines which message it is in the handshake
  - **replay counter** – used to detect replayed frames, replay counter will be increased
  - **nonce** - nonces (of supplicant and authenticator) to generate new keys
  - **RSC** Receive Sequence Counter - starting packet number of a group key
  - **MIC** - contains Message Integrity Check created with KCK
  - **Key Data** - contains group key encrypted with KEK
- encryption schemes used: AES-CCMP, GCM , MAC: Michael (weak), GHASH (from GCM)

handshake:

- notation: after ";" the data are encrypted
- green background = "sometimes"
- Enc_K^i is encryption with key K and IV i

association stage	supplicant		authenticator
		Authentication request $\rightarrow$	
		$\leftarrow$ Authentication response	
4-way handshake		$\leftarrow$ Msg1(r,Anonce)	
	derive PTK		
		Msg2(r,Snonce) $\rightarrow$	
		$\leftarrow$ Msg3(r+1,GTK)	
			derivePTK
		Msg4(r+1) $\rightarrow$	
	install PTK		install PTK
group key		$\leftarrow \mathrm{Enc}^{x}_{\mathrm{PTK}}(\mathrm{Group1}(r+2;\mathrm{GTK}))$	
handshake		$\mathrm{Enc}^{y}_{\mathrm{PTK}}(\mathrm{Group2}(r+2)) \rightarrow$	
	install GTK		install GTK

Table 1.

- state automaton definded, states for the supplicant:

  A) PTK-INIT:

  - entered when 4 way handshake started
  - exit to state PTK-START with Msg1 received
  - operations: PMK- preshared master key

  B) PTK-START:

  - exit: self loop with MSg1 received, with proper Msg3 to state PTK-NEGOTIATING (proper= MIC correct and no replay)
  - operations:
    - TPTK=CalcPTK(PMK,ANonce,SNonce)

  - Send Msg2(SNonce)

C) PTK-NEGOTIATING:

- exit: unconditional to PTK-DONE
- operations:
  - PTK=TPTK
  - Send Msg4

D) PTK-DONE:

- exit: to PTK-START if Msg1 received, to PTK-NEGOTIATING if proper Msg3 received

attack 1 - plaintext retransmission of Msg3

supplicant		adv		authenticator
	$\leftarrow$ Msg1(r,Anonce)		$\leftarrow$ Msg1(r,Anonce)	
derive PTK				
	Msg2(r,Snonce) $\rightarrow$		Msg2(r,Snonce) $\rightarrow$	
	$\leftarrow$ Msg3(r+1;GTK)		$\leftarrow$ Msg3(r+1;GTK)	
				derivePTK
	Msg4(r+1) $\rightarrow$			
install PTK				
	$\mathrm{Enc}^1_{\mathrm{PTK}}\{\mathrm{Data}(A....)\} \rightarrow$			
	$\leftarrow$ Msg3(r+2;GTK)		$\leftarrow$ Msg3(r+2;GTK)	
	$\mathrm{Enc}^2_{\mathrm{PTK}}\{\mathrm{Msg4}(r+1)\} \rightarrow$			
resinstall PTK				
			$\mathrm{Enc}^2_{\mathrm{PTK}}\{\mathrm{Msg4}(r+1)\} \rightarrow$	(rejected)
			Msg4(r+1) $\rightarrow$	
				install PTK
	$\mathrm{Enc}^1_{\mathrm{PTK}}\{\mathrm{Data}(B....)\} \rightarrow$		$\mathrm{Enc}^1_{\mathrm{PTK}}\{\mathrm{Data}(....)\} \rightarrow$	

mechanism:

- according to the 802.11 standard Msg4(r+1) will be accepted as it is checked that r+1 is a replay counter used before
- the problem is that $\mathrm{Enc}^1_{\mathrm{PTK}}\{\mathrm{Data}(A....)\}$ and $\mathrm{Enc}^1_{\mathrm{PTK}}\{\mathrm{Data}(B....)\}$ use the same IV but security of the encryption modes used collapse in this case