WIRELESS SECURITY: STANDARDS, SECURITY FLAWS and OPEN ISSUES.

Summary

- ✓ Introduction
  - The emergence of wireless;
  - The wireless Ad Hoc network paradigma;
  - Security challenges,
- ✓ Standards
  - Standards;
  - Weaknesses of a standard: Case Study;
  - Lesson learning,

Summary

- ✓ Open issues: re-keying
- ✓ Conclusions

References.

The emergence of Wireless (1/)

According to Gartner Group

"… corporations will be using wireless networks to extend mission-critical applications to mobile users."

The emergence of Wireless (2/)

- ✓ The segment of interest and today's use:
  - Private: point to point connectivity (e.g. Cellular telephony);
  - Business: reduce cycle time (introduce independence from the operator location);
  - Public sector: rescue operation (e.g. after air crash), police forces.

The emergence of Wireless (3/)

- ✓ The segment of interest and tomorrow's use:
  - Private: intra-devices integrated connectivity (home remote control);
  - Business:
    - business processes integration (vertical integration of the cycle//home-office convergence);
    - Assessment of exploitation in severe environmental condition (e.g. Ocean bed),

The emergence of Wireless (4/)

- Public sector: e-government enabling (e.g. Paper-based bureaucracy reduction);

- Military:
  - tactical communications (e.g. squad coordination);
  - Equipment check-up (e.g. Weapons and shield status) ;
  - Unattended surveillance (e.g. sealing);
  - Information gathering (e.g. NBC contamination, tank movement).

The Ad Hoc Networking paradigma (1/)

- ✓ An *Ad Hoc* network is a collection of nodes **not relying on a predefined infrastructure** to keep the network connected;
- ✓ Ad Hoc network can be formed, merged or partitioned on the fly;
- ✓ Nodes are often Mobile.

The Ad Hoc Networking paradigma (2/)

- Point of Strength:
  - completely distributed architecture;
  - virtually no single point of failure;
  - highly redundant;
  - handled/wearable devices;
  - low cost device.

The Ad Hoc Networking paradigma (3/)

- Point of weakness (1/2)
  - battery powered;
  - hardware constrained (e.g.small amount of RAM and Disk space available);
  - complex management of the *volatility* (e.g. Nodes can join & leave an Ad hoc network at high rate);
  - need of a set up phase;

The Ad Hoc Networking paradigma (4/)

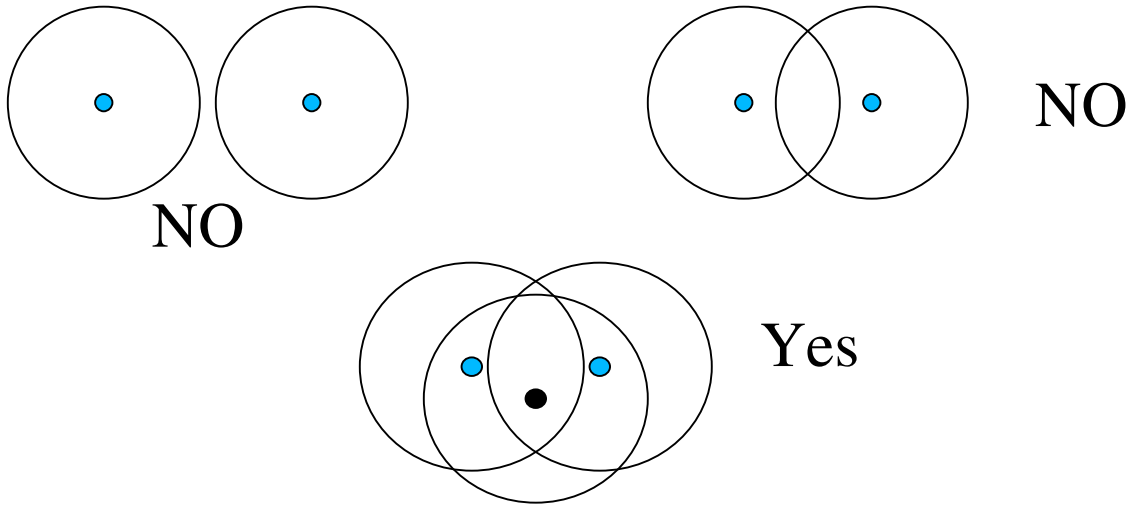
- Point of weakness (2/2)

  - routing complexity;
  - superimposing and maintaining a logical hierarchy over the network (while for industry reasons nodes are built equal);
  - limited communication range.

The Ad Hoc Networking paradigma (5/)

- **Communication range, Coverage area and routing (1/)**

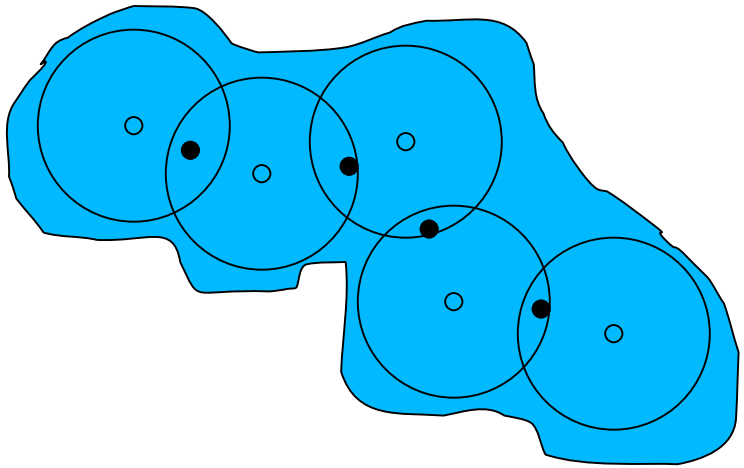
  - communication range: its importance in direct communication and coverage area.



The Ad Hoc Networking paradigma (6/)

- **Communication range, Coverage area and routing (2/)**

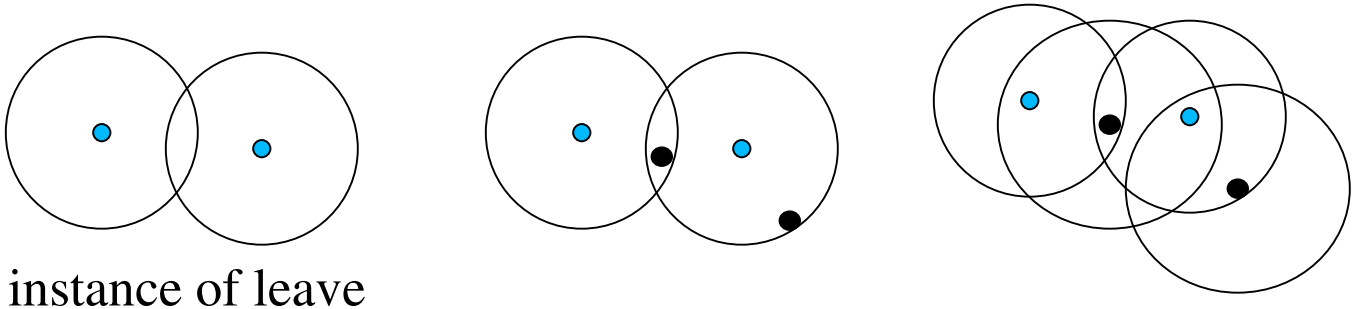
  - through routing the coverage area is only loosely coupled to the communication range



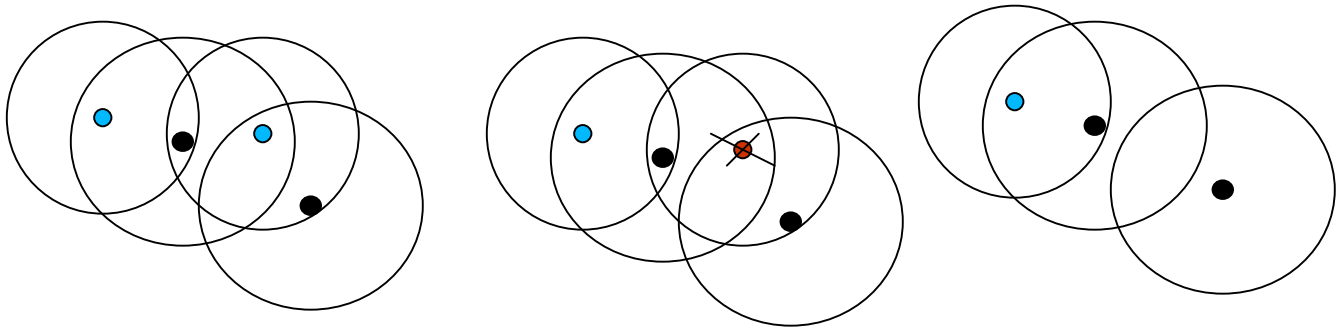
The Ad Hoc Networking paradigma (7/)

- Join, leave, split & merge

  - an instance of join



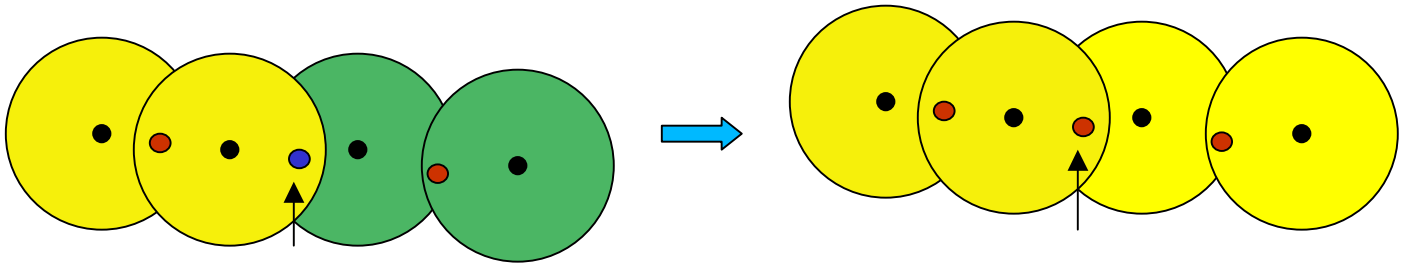
  - an instance of leave



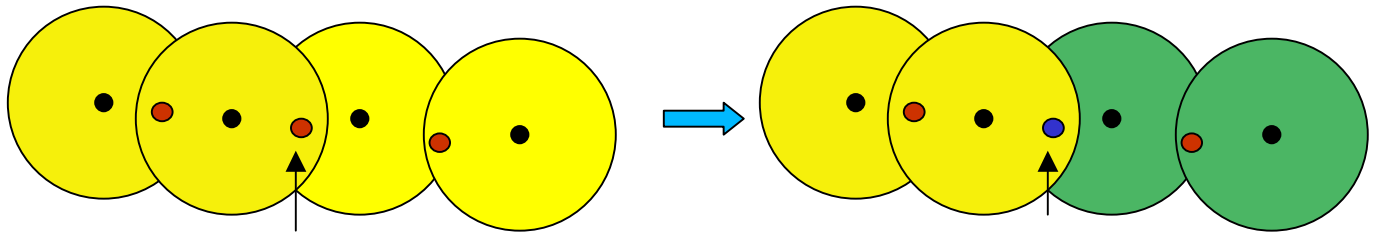
The Ad Hoc Networking paradigma (8/)

- Join, leave, split & merge

  - an instance of merge



  - an instance of split



Security challenges (1/)

The goal is the same than in the wired paradigma, i.e. to achieve (at least):

- **Confidentiality;**
- **Integrity;**
- **Availability.**

But... we can't pretend to start from the solutions in the wired paradigma and apply those solutions in the wireless context: the environment is quite different.

Security challenges (2/)

- ✓ Wireless Vs. Wired: there is no need to physical access: communication spreads in the air;
- ✓ Wireless Vs. Wired: resource constrained
  - Battery constrained;
  - Low processing power;
  - Small amount of RAM;
  - Small amount of disk space.

Security challenges (3/)

- ✓ Constrains on the Hardware have a fall out on the application level:
  - No expensive computing activities (e.g. no asymmetric cryptography - CPU is highly battery consuming);
  - No large databases on disk;
  - No correspondence tables in RAM,
- ✓ Adaptive routing: due to the highly dynamic topology ;
- ✓ No single access point: authentication is harder.

Security challenges (4/)

- ✓ Set up of the Ad Hoc infrastructure;
- ✓ Routing:
  - Data traffic;
  - System traffic,
- ✓ System management activities, such as:
  - Join and Leave;
  - Split and Pool,
- ✓ Application level: can introduce security flaws in itself, even if underlying levels are sound.

Security challenges (5/)

In short: security in Ad Hoc Wireless network is harder than in the Wired paradigma.

Standards (1/)

Commercial Standards

- IEEE 802.11: has a distributed media access control so it supports a virtually unlimited number of nodes in the same network, with up to 20 nodes talking concurrently.

- Bluetooth: point to multipoint technology using a centralised access control scheme with a master controlling the time division duplex traffic in the so-called *piconet*. The number of active concurrent nodes is limited to 8.

Standards (2/)

- Cut Off The Shelve

US Robotics Wireless

802.11 compliant

PCMCIA card

Euro 30.00

3Com Wireless

Bluetooth compliant

PCMCIA card

Euro 50.00

Standards (3/)

- Comparison

	Bluetooth	***802.11b***
Frequency	2.4 GHz	2.4 GHz
Data rate	1 Mbps	11 Mbps
Voice Channel	Yes	No
Range	10 m.	Up to 150 m.
Line-of-sight req.	No	No

Note: Bluetooth is intended for home deployment, while 802.11 is also known as Wireless LAN.

Standards (4/)

- under investigation (technical assessment)
  - 802.15 Working Group -WG- WPAN (wireless personal area network)
    - P802.15 Task Group -TG- (WPAN based on Bluetooth V1.0 spec.);
    - P802.15 High Rate Study Group (obj.: to develop a physical layer and a MAC layer for high rate, low complexity, low power consumption wireless connectivity)

Weaknesses of Standards: Case Study (1/)

- ✓ IEEE 802.11 security is based on the Wired Equivalence Privacy Protocol (WEP) -i.e. bring the security level of the wireless environment close to that of wired ones-
- ✓ In particular, we will address two points:
  - **C**onfidentiality: showing how messages can be eavesdropped;
  - **I**ntegrity: showing how it is possible to tamper a message.

Weaknesses of Standards: Case Study (2/)

LOGICAL FRAMEWORK OF WEP

- ✓ The **sender** set up:
  1. given a message (M), we compute a checksum of M (c(M)) and then concatenate the checksum to M, i.e. <M, c(M)>. Note: the checksum employed is the CRC;
     1. Let P=<M, c(M)>;
  2. We encrypt P using RC4, that is, we choose an **i**nitialisation **v**ector (v) and a key k, than we compute RC4(v,k) and xor it with P;
     1. Let C= (P xor RC4(v,k)).

Weaknesses of Standards: Case Study (3/)

- ✓ Then -node A- transmit to -node B-:

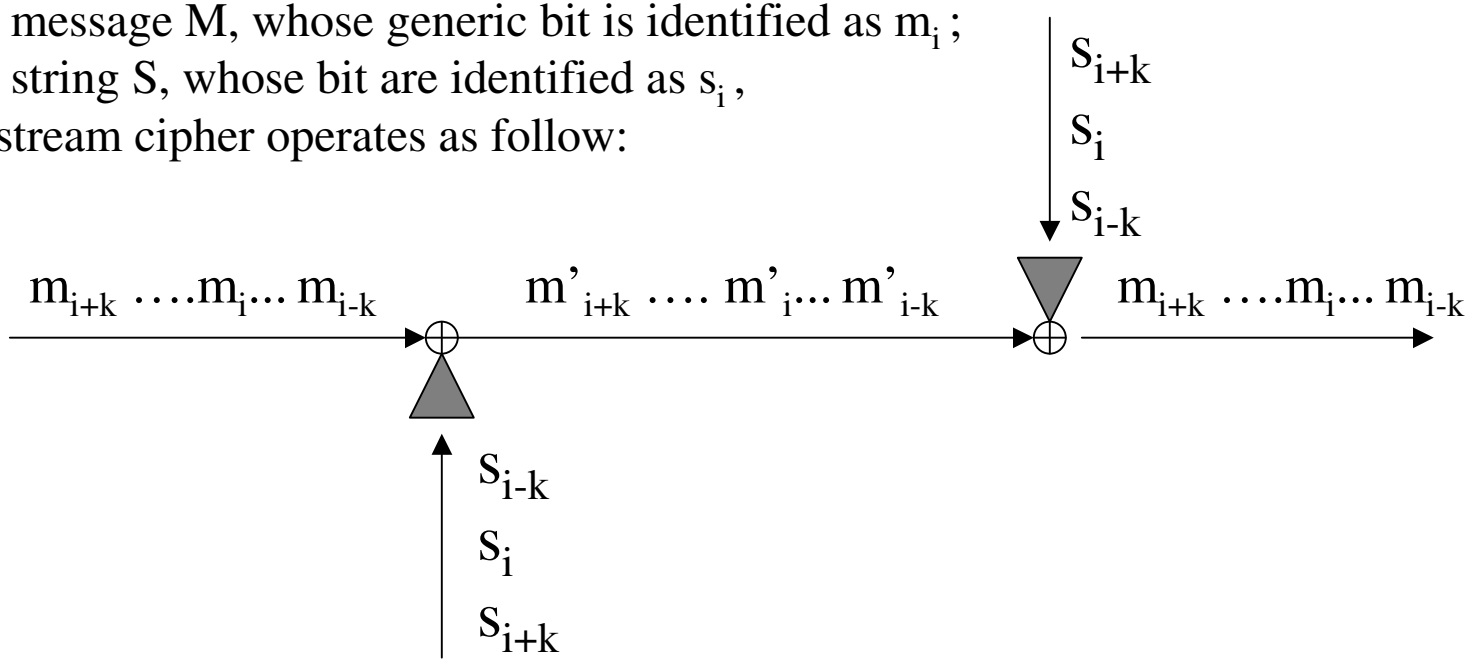
  A → B: [v, C] -i.e. [v, (<M, c(M)> ⊕RC4(v,k)]-

- ✓ The **receiver** will compute:
  - *RC4(v,k)*;
  - 🗁 <*M, c*(M)>= C⊕RC4(*v*,*k*)
  - 🗁 *c' = Checksum (M)*
  - 🗁 *Compare c' and c*(M): if the two string matches than accept the message, otherwise discard (integrity has been corrupted).

RC4: an OFB Stream Cipher

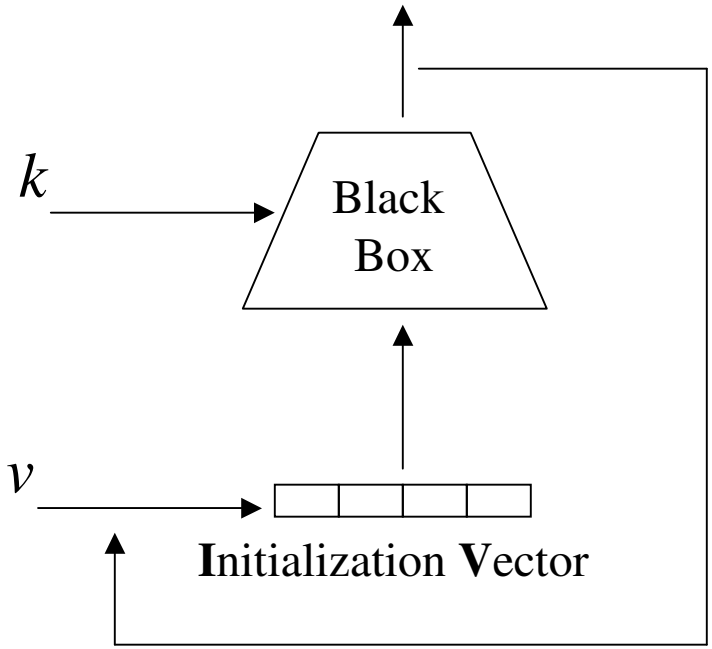
1. Stream Cipher

Given:
- a message M, whose generic bit is identified as m_i ;
- a string S, whose bit are identified as s_i ,
 a stream cipher operates as follow:



RC4: an OFB Stream Cipher

2. Output Feedback Mode (OFB)



Note: the Black Box for RC4® is a trade mark, but actually it is public domain

Weaknesses of Standards: Case Study (4/)

- ✓ Mining Confidentiality (1/)
  - ➢ fundamentals (1/2)
    - The RC4 stream cipher is the cornerstone for confidentiality in WEP.
    - If two messages are encrypted with the same *v* and *k*, information can be inferred on both messages:
      1. Given $C_1 = (P_1 \text{ xor } RC4(v,k))$ and $C_2 = (P_2 \text{ xor } RC4(v,k))$;
      - $C1 \text{ xor } C2 = (P_1 \text{ xor } RC4(v,k)) \text{ xor } (P_2 \text{ xor } RC4(v,k)) =$ $= P_1 \text{ xor } P_2 \text{ xor } (RC4(v,k) \text{ xor } RC4(v,k)) = P_1 \text{ xor } P_2$.

Weaknesses of Standards: Case Study (5/)

- ✓ Mining Confidentiality (2/)
  - ➢ fundamentals (2/2)
    - Thus, we have the XOR of two plain text (P_1 and P_2) and standard techniques can now apply (e.g. frequency analysis).
    - In particular, if n cyphertext reuse the same keystream, we have a problem of *depth n*. The bigger is n, the easier is the task to resume the messages.
    - However, note that *v* is in clear, but the secret *k* is unveiled, thus confidentiality *seems* guaranteed.

Weaknesses of Standards: Case Study (6/)

- ✓ Mining Confidentiality (3/)
  - ➢ Gathering information about *v*
    - ☙ The WEP **standard**:
      - recommends to change *v* at any transmission (but **it is not** mandatory);
      - It does not specify how to change *v*;
      - Fix *v* as only 24 bit long!
    - ☙ The WEP **implementation**:
      - in many implementations, at any start up of the PCMCIA *v* starts from a fixed value – 0 –;
      1. at a successive transmission, *v* is simple augmented with step of length 1.

Weaknesses of Standards: Case Study (11/)

- ✓ Mining Integrity (1/) :
  - ➢ Exploiting *crc* (1/)
    - ☙ We will show how to arbitrarily modify M, without detection

      Assumption:

      the checksum (*CRC*) is a liner function of the message, i.e. *CRC*(M xor M') = (*CRC*(M)) xor (*CRC*(M'));

      Thesis:

      it is possible to find a new cipher text M' such that: M'= (M xor Δ) where Δ is arbitrarily chosen (without knowledge of M).

      Note that, unless we know the plain text, the result of such transformation is **meaningless**, Nevertheless, the receiver cannot distinguish forged message from an original one.

Weaknesses of Standards: Case Study (12/)

- ✓ Mining Integrity (2/) :
  - ➢ Exploiting *crc* (2/)
    - we intercept a message C = [v, (RC4(v,k) xor <M, c(M)>];
    - Chosen a delta, we prove that the message will be correctly received as M' where M'=M xor Δ.
    - Proof:
      - we compute CRC(Δ); than pose C'= C xor (Δ, CRC(Δ));
      - C'= C xor (Δ, CRC(Δ))= ((RC4(v,k) xor <M, c(M)>) xor (Δ, CRC(Δ)) = RC4(v,k) xor **((<M, *c*(M)>) xor (Δ, *CRC*(Δ))=**
        =RC4(v,k) xor **((<M xor Δ, *c*(M) xor CRC(Δ)>)=** RC4(v,k) xor <M', c(M')>;
    - So C' will be correctly decrypted by the receiver as M' (i.e. M' is recovered by decryption and its *checksum* matches CRC(M')).

Weaknesses of Standards: Case Study (13/)

- ✓ Mining Integrity (3/) :
  - ➢ message injection (1/)
    - ☙ As shown in the previous example, the checksum is unrelated to the key!
    - ☙ Thus, given a plaintext, the *CRC* can be trivially computed.
    - ☙ Therefore, once recovered a plain text and a cypher text, the keystream can be easily obtained through xoring, i.e.
      - ❿ Having eavesdropped ((RC4(v,k) xor <M, c(M)>);
      - ❿ recovered M (by one of the previously exposed techniques);
      - ❿ computed *CRC*(M),

Weaknesses of Standards: Case Study (14/)

- ✓ Mining Integrity (4/) :
  - ➢ message injection (2/)
    - ☙ we obtain: ((RC4(v,k) xor <M, c(M)>) xor <M, c(M)> = RC4(v,k) xor (<M, c(M)> xor <M, c(M)>) = RC4(v,k) xor 0_n = **RC4(v,k)**;
    - ☙ thus, if we want to inject a message M' (this time the message can be meaningful) we have only to compute:
      - ❿ RC4(v,k) as previously exposed;
      - ❿ *CRC*(M');
      - ❿ send [v, ((RC4(v,k) xor <M', c(M')>)] - v is in clear-.

Weaknesses of Standards: Case Study (15/)

- ✓ Mining Integrity (5/) :
  - ➢ message injection (3/)
    - ☙ the IEEE 802.11 standard recommend, but does not impose to change *v* after each transmission, thus:
      - ❿ *v* is in clear;
      - ❿ once we know RC4(*v*,*k*);
      - ❿ we can inject messages at will, employing the same *v* and RC4(*v*,*k*) !;
      - ❿ the receiver MUST accept such messages.
    - ☙ the fact that a receiver must accept a message with an already used *v*, is due to standard: if it would not accept such a *v*, it would not be standard-compliant!

Weaknesses of Standards: Case Study (16/)

- ✓ Everything works, under the assumption that the key in RC4(v,$\underline{\boldsymbol{k}}$) stays unchanged. Is it reasonable?
  - ➢ Key management: specification of the **standard**
    - ☙ the standard does not specify how key distribution is accomplished: it relies on an external mechanism;
    - ☙ we can choose the key from an array of 4 keys (the message carries the identifier of the position of the key in the array),
  - ➢ Key management: **implementation**
    - ☙ a single key is often used;
    - ☙ key refresh is not frequently performed (*it implies reconfiguring network drivers*).

So, the answer is: yes, it is.

Lesson learning (1/2)

- To do:
  - The **standard is open,** so further enhancements are possible: standards, also commercial would be-standards, should be publicly reviewed before adoption, otherwise a successively unveiled security flaw will vanish million $ of investment;
  - Adopt a correct **Methodology** to address security points (confidentiality, access control, data integrity);
  - **Rely** on already developed solution (but... see next).

Lesson learning (2/2)

Not to do:

- Don't assume today's technology limits as a supporting point for security (tomorrow's technology -two years time- will overcome those limits) -e.g. expensive technical equipment-;
- Don't re-invent the wheel: base development on already well known protocols and standards (adopt, as far as possible, already scrutinised solutions -e.g. IPsec suit-);
- Don't rely on already developed solutions that have already shown weaknesses (e.g. adopt a *CRC* to act as *MAC -MD5, SHA-*).

Open issue (1/)

- One of the main security concern arises from the unavailability of asymmetric cryptography;
- widespread viable solution is to resort to symmetric cryptography;
- Draw back: **key management.**

Open issues (2/)

- Naïve considerations on symmetric key management:
  - If each node holds a different key:

    If *n* actors need to communicate

    ⇓

    n*(n-1) key exchange must occur.

    This implies:
    - Continuos exchange of keys among nodes;
    - resource consumption.

  - On the opposite side, if only a key is employed if a node is compromised all the network is compromised.

Open issues (3/)

- Constraints:
  - avoiding single point of centralisation (both for security and capacity reason);
  - partitioning the key space to:
    - enforce multilevel security;
    - enable specialisation of nodes,
  - key refreshment, to avoid cryptoanalysis based attack to succeed.

Open issues (4/)

- Key management: possible choices
  - **Centralised** approach: only one group controller;
  - **Distributed subgroup** approach: the management of group is divided among subgroup managers;
  - **Distributed** approach: there is no explicit group controller and the key distribution is done by the members themselves.

Open issues (5/)

- **Key management possible choices: Pro$_s$ and Con$_s$**
  - **Centralised** approach:
    - single point of failure (-);
    - excessive burden on a single node (-);
    - facilitate management (+);
    - enforce security (+).

    Despite its points of strength, the first two Con$_s$ are blocking point for the adoption of such a choice (nowadays unfeasible)

Open issues (6/)

- **Key management possible choices: Pro$_s$ and Con$_s$**
  - **Distributed subgroup** approach:
    - enforce cluster partitioning (+);
    - security flaws in as subgroup does not reflect to the network as a whole (+);
    - enable specialisation of sets of nodes (+);
    - allow a certain degree of decupling (+)

Open issues (7/)

- **Key management possible choices: Pro$_s$ and Con$_s$**
  - **Distributed** approach:
    - need the super-imposition of a logical infrastructure (-);
    - security flaw in a node implies:
      - .. the failure of the whole network if all nodes are equal (-);
      - .. the failure of the single if nodes are different (+),
    - highly available (+);
    - completely decupled (+).

Open issues (8/)

- Degrees of freedom:
  - triggering of key refreshment:
    - time triggered: ;
    - event triggered: ;
  - scope of partitioning: to what extent multilevel security must apply;
  - degree and extent of node specialisation (e.g. routing, data and key management nodes).

Open issues (9/)

- A few Hard challenges:
  - authentication of:
    - newly joined nodes;
    - sleeping node waking up or re-joining;
  - management of **non blind trust** among peer;
  - Intrusion detection monitoring and signalling;
  - …
  - …

Conclusions (1/)

- Security challenges in ad hoc wireless network:
  - original environment and application;
  - resource constrained;
  - probably currently unveiled security problems will rise as the technology spreads.
- Security problems in 802.11 standard:
  - good example of how to crack a standard;
  - exploits ready to use: (smart) implementation will follow;
  - challenge to the industry: defence is on the move.

Conclusions (2/)

Re-keying

- good research field because:
  - multiple degrees of freedom;
  - tight constraints,
- the two competing approaches (i.e. distributed vs sub group distributes) are probably application-class bounded in their scalability and efficiency;
- a powerful and useful means to implement:
  - multilevel security;
  - network security management (join, leave, split ...);
  - a secure application framework.

References

- "Securing Ad Hoc Networks"; Lidong Zhou and Zygmunt J. Haas; IEEE Networks, Vol. 13, issue 6, Nov-Dec 1999; pages 24-30;
- "Security in Ad Hoc Networks"; Vesa Kärpijoki, Helsinki University of Technology, Internet draft;
- "Intercepting mobile communications: the insecurity of 802.11"; Nikita Borisov et al; The 7th Intl. Con. On Mobile Computing and Networking 2001, pages 180-189;
- "An inductive chosen plaintext attack against WEP", W.A. Arbaugh, IEEE Document 802.11/203, May 2001;
- "Kronos: a scalable Group Re-Keying approach for secure Multicast", Sanjeev Setia et al., IEEE Symposium on security and privacy, Oakland CA, may 2000;
- "A Decentralised Architecture for Group key management", Sandro Rafaeli, Lancaster University, Ph.D. thesis