Security of Wireless Communication

Dan Cvrcek

Brno University of Technology
dancvrcek@gmail.com *

Abstract. What are the real security issues of wireless communication and wireless sensor networks in particular? Despite predictions of wireless sensor networks being deployed in many aspects of everyday life, real world deployments are still quite sparse. It seems that monitoring of large civil engineering structures is one of the few applications where wireless sensor networks may give enough value for the necessary investment. The least, several companies managing large civil structures in the UK are keen on investigating the potential of wireless sensor networks.
In the light of this technology, which is built on a new paradigm of dense wireless communication networks, we can see new security challenges never experienced by engineers before. Can we appreciate the difference between wire and wireless communication and also the difference between centralised wireless networks, e.g., WiFi and largely decentralised sensor networks? We show how the shift in the technology introduces new problems that need to be solved to provide secure communication systems. The second part of the paper details particular attacks that work against current implementations of wireless sensor networks and routing, traffic analysis, and cryptography in particular.

1 Introduction

The history of wireless and wired communication intertwines. We have used wireless optical communication systems until the nineteenth century when electricity was discovered and we learnt that it was possible to send sound and signals through a wire. Lengths of communication links increased largely when the voice was replaced with the Morse code. Marconi was behind first practical radios able to send electrical signals over the air at the beginning of the twentieth century. He increased the radio range enough to allow sending messages across the Atlantic ocean.

The wireless communication was cheap but it did not allow to connect two persons willing to speak to each other. The first telephone systems were wired and thanks to the networks built in the early years of telephoning, we still use land-line telephones. The wireless technology is more complicated but it has recently become reliable and cheap to compete with, and possibly replace, wired systems in certain scenarios.

* The paper is based on the work carried out with Frank Stajano and Matt Lewis as part of the EPSRC WINES Infrastructures project at the University of Cambridge, UK.

For Wire	Against Wire
Well defined transmission medium	Cost of the infrastructure
More options for network management	Ownership of links between nodes
Limited interference	Fixed infrastructure
High bandwidth	

Table 1. What is the wire communication about.

What are actually the advantages of wire communication? When you look at Table 1 you may realise that wired communication is more suitable for networks featuring a large number of nodes with many connections. The more the network changes into a sparse graph with long links, the cons gain on the importance. As a matter of fact, we can list specific scenarios, where wireless technology dominates:

- fast-changing topologies – GSM and WiFi;
- sparse network topologies – Microwave links, WiFi;
- short-range communication – Blue-tooth, ZigBee; and
- low-cost, quick network deployment – ZigBee, WiFi.

In any of these scenarios, wireless communication will be the preferred technology and the economic advantage further increases in locations lacking an existing wired, land-line network.

There is one more low-point of wireless communication. It is restricted by regulations of the public frequency spectrum use. There are very few frequency bands available for digital communication systems and they cover frequencies from about 1 to 5 GHz. As a result, the power of transmitters is strictly regulated and the communication distance is limited so that neighbouring transmitters do not interfere with each other.

However, this holds only for "legitimate" networks. Attackers would not feel to be bound by the limitations and not only because they usually stay in one place too shortly to be caught. The transmitting power of adversaries much higher than that of legitimate users is only one of the aspects underlining security challenges for wireless communications.

2 Security Problems

Due to omnidirectional transmission, there are three main security subjects differentiating wireless from the wired communication:

- Authentication / masquerading – robust authentication of the other end of the communication channel;
- Relaying – ensuring that the communication is happening in real time and is not maliciously delayed by a whatever small amount of time; and
- Eavesdropping – ensuring that no one can listen to the communication without being authorised or detected.

2.1 Authentication

Security of any security protocol depends on the assumptions stated for a given system. It is not possible to say whether a protocol is or is not secure until someone defines what is meant by "secure". Everyone knows that the Needham-Schroeder protocol [1] is broken. When one reads the paper, it seems that its authors assumed there are two sets of users, legitimate and attackers, and that the legitimate users were not supposed to attack each other. This is not explicitly stated though and the protocol gets broken only after this assumption is ignored or removed.

We believe that most of you are familiar with the GSM technology. Any communication channel (a connected call) consists of three logical parts: a wireless connection between the caller and a Base Transceiver Station (BTS), a wireless connection between another BTS and a callee, and a back-end wired leg connecting the two BTSes. No one has been really much interested in the security of the middle leg as there is no cryptography deployed and any attack is possible so long as one can get access to the wire. However, a lot of cryptographic research has been carried out for the wireless links.

There are two cryptographic algorithms – A3 and A5 – providing cryptographic assurance that no unauthorised person can eavesdrop on calls or masquerade and initiate or accept calls on someone else's behalf [2, 3]. We know today that the algorithms are cryptographically weak but any attack still needs a lot of mathematics and special equipment or software.

It is much less widely known that the GSM protocol suite is also broken because it does not require two-way authentication. The BTS stations do not authenticate themselves. GSM standards only require users (their handsets) to authenticate to a BTS. Is it possible for someone to masquerade as a BTS and accept calls from / to users in their communication range?

This attack does not require breaking any cryptographic protocol but one needs a special equipment that is hard to get by – not even on eBay. One needs a special licence and only mobile phone operators or specialised agencies are able to acquire it. A BTS is also a quite expensive piece of equipment to buy.

This has however changed recently with GSM Femtocells. Vendors of home and small business network routers realised that there may be a demand for devices forwarding GSM phone calls to VoIP systems, e.g., Skype and 3G data connections to a cable broadband connection. It would also solve a problem of weak GSM signal in some buildings. As a result, wireless routers with interfaces for GSM, ADSL and WLAN were introduced with a quite affordable price tag of about twice as much as for WiFi home routers.

Figure 1 shows communication ranges for different types of GSM cells. The range is limited by the antenna used on the cell's base station. A restriction quite easy to overcome. (Attackers usually do not feel to be bound by limits imposed on transmission power by regulators.)

Thanks to the advance in the GSM technology, man in the middle or impersonation attacks are now within reach of attackers with shoestring budgets.

Cell type	Typical cell size	Data rate limitation
Macro	1 – 30 km	Propagation
Micro	200m – 2km	Capacity and propagation
Pico	4 – 200 m	Capacity and propagation
Femto	10 m	Broadband connection and handset

Fig. 1. Types of GSM cells.

2.2 Relaying and eavesdropping

Wireless technology is susceptible to relay attacks when the attacker creates a transparent tunnel between a sender and a recipient. Attacks on communication between RFID cards and readers are typical examples studied in several papers (e.g. [4, 5]).

The problem of the RFID technology is that it was designed to remove interventions from users. Any RFID card will start an authentication process whenever it is placed in the proximity of a reader allowing for opening doors or paying for lunch in a canteen without removing the card from a wallet or even pocket.

The security was deemed to be sufficient as the communication range of RFID cards is less than 4 inches and either the card authentication enabled only low value transactions or there have been other security mechanisms in place providing an additional layer of security.

Four inches, is it really the maximum distance? Gerhard Hancke et al. [4] conducted a thorough research of RFID capabilities and studied two scenarios, for a passive and an active attacker.

1. Passive attacker – the attacker is only trying to eavesdrop on messages sent from a card to a reader. This scenario is applicable on situations when cards use a static response for their authentication or when they send sensitive data to readers, e.g., personal information sent by a passport at customs. Authors were able to optimise the antenna and increased the possible distance from an RFID card to the antenna to 4 meters.
2. Active attacker – attacks in this scenario try to increase the distance between a reader and a card by using a stronger electromagnetic field generated by an improved antenna. They were able to increase the communication range to 1.5 meter, while the reader was 15 cm away from the smart-card.

Communication range is quite an interesting topic. John Hering developed a blue-tooth rifle in 2004-5 [6]. The maximum communication range of Bluetooth devices is well below 50 meters, typically 10 meters. John's gun was able to tap bluetooth devices (e.g., perform a passive attack) from over a mile away, during experiments carried out against devices in high office buildings in New York.

Increasing communication range improves attackers' ability to communicate with a card (or other wireless device). They can then use the device as an oracle

to authenticate transactions taking place even kilometres away from the card by relaying the card responses via a WiFi or a low-delay wireless connection. RFID standards and implementations introduce maximum delays, but they are very generous in terms of maximum distance available for relay attacks.

3 Wireless sensor networks

Wireless sensor networks represent just a small fraction of wireless networks but they abstract some of interesting new concepts in distributed computing and their existing practical implementations re-introduce security challenges of wired communications in a very different environment.

There is an abstraction of sensor motes called smart dust. Smart dust represents tiny motes (just a few square millimeters), powered by a battery or sollar energy, and very cheap to produce. It is also possible, in this abstraction, to deploy tens of thousands of motes in a single network.

There has been published a large body of theoretical research into properties of wireless sensor networks – very large networks of very simple nodes (motes). Such networks were presumed to be deployed in large batches (e.g., by throwing them off a plane) followed by a self-organising phase, automatically and autonomously launched after the physical deployment of the motes. The large quantity of motes brings in practical constraints: it is expensive to "personalise" motes by changing the code or data stored on the motes. It is much easier to mass-produce sensors that are identical even on firmware and configuration level.

A lot of security research has been devoted to key management schemes in this special environment and particularly to key pre-distribution schemes. Key pre-distribution schemes expect any two nodes to establish a shared pairwise (link) key when they happen to be physical neighbours after their deployment. As sensor networks are assumed to form dense graphs, the probability of two randomly selected nodes sharing a common key can be much lower than 100 %. Theoretical models based on this assumption introduce a trade-off between the network connectivity and the memory required to store keys on nodes.

The idea of random key pre-distribution for wireless sensor networks was firstly introduced in [7] as the EG scheme. Here, each node contains a random subset of keys from a large set of keys. Motes perform a key setup phase identifying subsets of shared keys between any two neighbours and these keys are subsequently used to secure communication between the two motes. It is possible to use probability theory to compute ideal sizes of key sets to ensure connectivity of large and dense networks. There are various extensions of this scheme. Authors of [8] introduce a scheme requiring at least q shared keys instead of one. Another approach uses pseudo-random generation, instead of random selection, of key indexes [9].

Pairwise key pre-distribution is another scheme. Any given key is shared by exactly two nodes in a network and a compromise of any mote does not compromise any other mote in the network. As opposed to schemes in the previous

paragraph, where capturing of a very small subset of network motes may reveal a majority of keys used in the network.

4 Real wireless sensor networks

The following sections describe practical security issues one encounters when commercial off-the-shelf wireless sensor nodes are to be used. The first thing we have to mention is that the networks are much different from what has been described in the previous section.

There areseveral vendors of general purpose wireless sensor kits, although it seems that academic research is still the main market. Most widely used platform is TinyOS developed as a GNU project. Several commercial products, including Xbow we worked with, are extensions of TinyOS. The presented results are not theoretical results but outcomes of experiments with real implementations of sensor networks.

4.1 Typical deployment

We experimented with mesh networks built from MICAz motes with mounted sensors designed by civil engineers. These motes run TinyOS system and wireless communication is implemented with IEEE 802.15.4 compliant radio chips. The standard defines maximum link bandwidth to be 250 kbps.

Battery life of motes might provide several years of up-time if the communication was initiated once a day. Current setup introduces communication several times a minute and batteries last for 4-5 weeks. We have built improved nodes with special D and DD size batteries that should last more than a year.

Our mesh networks consist of clusters of 10-30 motes connected to one relaying gateway. These clusters, including the gateway, are independent of each other without any direct connections. The clusters connect via their gateways to a central computer managing the networks and collecting data.

Gateways are Linux boxes (Stargate [10]) with one MICAz mote for a 802.15.4 / ZigBee connection to the mesh network. This mote talks to its gateway via an RS-232 interface. Gateways connect to the central computer using a WiFi router with a GPRS module or an ADSL router (particular technology depends on the physical location and available networks).

5 Attacker modelling

We tried to model an attacker before we started practical experiments. Practicality of attacks has been re-assessed after experiments to reflect difficulty of attack scenarios. We used two approaches to find out probable attackers. The first approach was to interview owners of large civil structures where we deployed the networks for monitoring to find out what would be the networks' use in a few years time and what they see as major risks. These interviews identified curious

hacker interested in the technology as the most likely attacker as the systems are not foreseen to provide any valuable data over short periods of time.

The second approach was to build a simple classification of attackers according to their knowledge and to the access to a sensor network they need to carry out certain attacks. Let us start with different types of access that may be needed for different attacks.

1. Remote access over the Internet – attacker may connect from anywhere and it is very hard to find them, identify them, or prosecute them.
2. Remote access over national/local communication infrastructure – attacker exploits access through infrastructures that are either local (WiFi networks), or with otherwise limited access from a certain area.
3. Physical proximity to system – attacker needs to get very close to the deployed network – distance in the range of tens of meters or less. It allows him to use communication means of motes or perform DoS attacks that require interaction with components of the network.
4. Physical access to single elements of system – attacker is able to physically touch particular motes, gateways of the network. This allows them tampering the device and re-program, replace or remove parts of the device or the device itself. The time and expertise may greatly vary (e.g., connecting to a mote would take a few seconds while reading out a permanent memory may require substantially more time).
5. Physical access to all (or most of) elements of the system – the most expensive scenario requiring attacker to get access to a large number of devices.

One can see that the first three options are achievable even for a low budget attacker. All our networks are connected to the Internet, some of them are even in publicly accessible areas. Physical access to networks deployed in underground systems (London underground in our case) is difficult and a physical proximity can be achieved easily only on a train – i.e. for very short time periods.

The second important issue is power of the attacker. This can be viewed from three different angles: money, knowledge, and personnel. One extreme is formed by an attacker without money, little knowledge and no personal (just him/her) – often called script kiddies. The opposite extreme is someone with unlimited money, detailed knowledge about the system and technologies being used, and of course enough personal to implement desired attack scenarios.

Attack scenarios we want to pursue assume attackers between the least powerful ones and a skillful hacker able to change the code for motes with limited budget that allows buying off-the-shelf products. We will also assume that the attacker can get access to the system as specified in the first three (or four in some cases) options from the list above. The most relevant types of threats are:

1. attacks on wireless communication:
   - eavesdropping communication;
   - analysis of gathered data; and
   - injection of new traffic (or replay attacks).

2. attacks exploiting decentralisation of the network management:
   - data communicated between sensors and sensor-gateway;
   - sources of data;
   - routing algorithms; and
   - how to defeat countermeasures when gateways / sensors check integrity of each other.
3. physical access to a device and subsequent:
   - changes in software/firmware;
   - spread of changes (infection) to other network nodes / gateways; and
   - disabling device.

6 Selective jamming – debugging mode

Jamming is definitely a low-cost attack. We implemented powerful jamming attacks without requiring any special hardware and based only on changes or extensions of available software. Such attacks are highly relevant as they allow for implementations by a relatively high number of potential adversaries.

As we were deploying MICAz motes in our monitoring networks, we chose MICAz (with the CC2420 radio chip) as the basic hardware platform for attack implementations. The motes are easy to buy and all necessary software is available as a freeware on the Internet. Micro-controllers on these motes are quite slow (clocked on less than 8 MHz) and re-implementation of the attack on almost any other platform will be undoubtedly feasible with respect to computational requirements.

Current implementation requires the criteria triggering the attack to be defined in advance. The criteria are in a form of matching conditions for selected bytes of messages and they are compiled into the code.

Once the mote is uploaded with the code and deployed, it keeps listening to the traffic. The mote eavesdrops enough bytes, decides whether the received content satisfies the pre-programmed criteria and if so, it switches the radio to Tx mode and jams the rest of the packet.

The most difficult step was to implement byte by byte listening. CC2420 chip normally receives an entire message (frame), stores it in a buffer, and raises a signal to the micro-controller to download the frame. When the micro-controller needs to transmit a packet, it uploads the whole packet to the internal buffer of the CC2420 and signals back that the content should be transmitted.

What we need for the attack to work, is the ability to listen to single bytes of the message and to stop listening at any time. Fortunately CC2420, as all other radio chips we have seen, features a debug mode that should be used for testing basic functionality of the chip. This mode allows single bits to be read by the micro-controller as they are received from the air. It means that we can do exactly what we want. The micro-controller takes care of the synchronisation, reading, and storing the data bits (fig. 2 shows a clock signal and first message bits provided by CC2420).

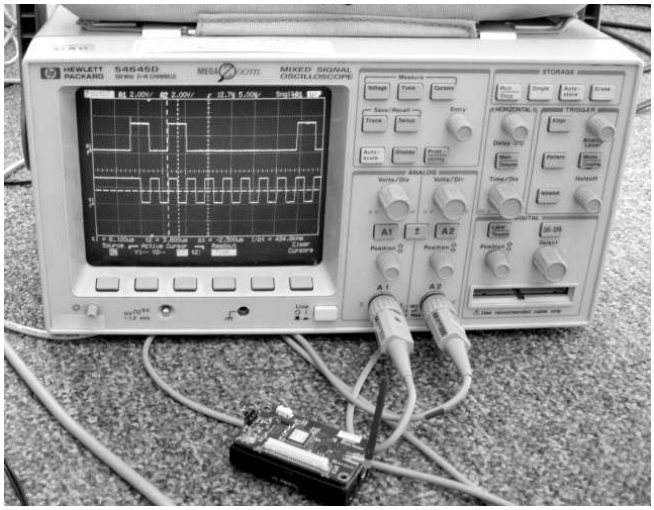


Fig. 2. Start of a frame – clock of the radio chip (top line) and sampled data bits (bottom line).

The application implemented in NesC language (a macro language based on C for TinyOS programming) not only correctly reads / eavesdrops messages, but it is also very code efficient.

6.1 Frame format

The frame format as used by MICAz motes differs from what was described in [11] as changes were introduced with the switch to the new radio chip.

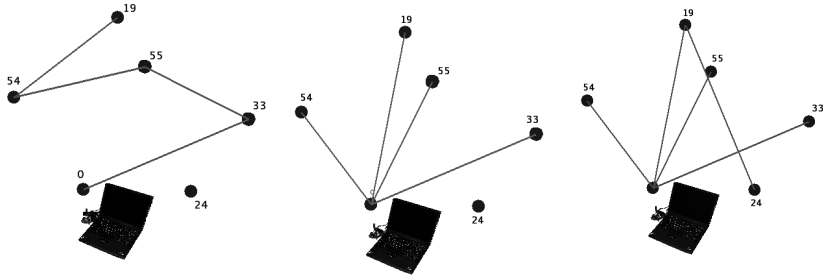
$$\begin{gathered} \textit{length}\ (1\text{B}) \mid fcf\ (2\text{B}) \mid \textit{dsn}\ (1\text{B}) \mid \textit{destpan}\ (2\text{B}) \mid \textit{Dest}\ (2\text{B}) \mid \\ \mid \textit{AM}\ (1\text{B}) \mid \textit{GrpID}\ (1\text{B}) \mid \textit{Data}\ (\leq 29\text{B}) \mid \textit{CRC}\ (2\text{B}) \end{gathered}$$

Items *length*, fcf, *dsn*, *destpan* are parts of 802.15.4 MAC layer. fcf (frame control field) says whether it is a data or some other type of frame. Destination mote address is of just two bytes (*Dest*). *dsn* is an eight bit serial number of the packet (used only to match acknowledge (ACK) frames confirming a frame reception with the original frame). *destpan* is always set to indicate broadcast ($0xFFFF$) to ensure that all motes will listen to all the messages. The remaining items in the depicted frame contain a TinyOS message itself.

TinyOS applications usually compile with multi-hop support. This functionality is based on a special seven bytes long routing header at the beginning of the *Data* field. It contains (*source address* (2B), *original address* (2B), *sequence number* (2B), and *hop count* (1B)). This would be followed by the data generated by the mote with the *original address* ID.

6.2 Jamming

The trigger condition we used was the *original address* to match a certain value. This allows jamming frames from selected motes anywhere in the network because *originaladdress* does not change. We did the first tests on a small network consisting of seven motes around the office. The topology of the network was a simple star as all the motes were able to directly reach the gateway (see Fig. 3).



a) jamming started b) stabilised topology c) jamming stopped

Fig. 3. A network during and after jamming of node 24.

The visualisation as showed in Fig. 3a) demonstrates immediate disconnection of the jammed node (mote with ID 24) and a short instability of the network topology when the jamming started. The topology has returned to the star shape after a very short time and the jammed node remained disconnected – Fig. 3b).

The second set of tests was based on jamming a mid-range connection (4 – 15 meters) between two motes, with different positions of the jamming mote. Overall efficiency was usually close to 100%, even for the jammer much further away, and in different directions from the receiver (see Figure 6.2). However, there were several occasions when the jamming was very ineffective, even in configurations that previously showed high success rates.

Electrical engineers told us that the anomalies are very likely to be caused by signal reflections in the particular environment. It may be therefore plausible to eliminate them with using a couple of jamming motes.

Despite this unpredictability in the test results, we believe that the attack is very powerful, and it constitutes a serious threat. The experiments, we have performed, used jammer with the same antenna and transmission power as were of other transmitters, but these can be easily replaced / increased.

6.3 Defences

We obviously can not eliminate jamming attacks completely. What we can do is to make it harder for adversaries to implement power efficient jamming attacks, and rebalance cost-benefit ratio of the attacks. Wood et al. analyse in

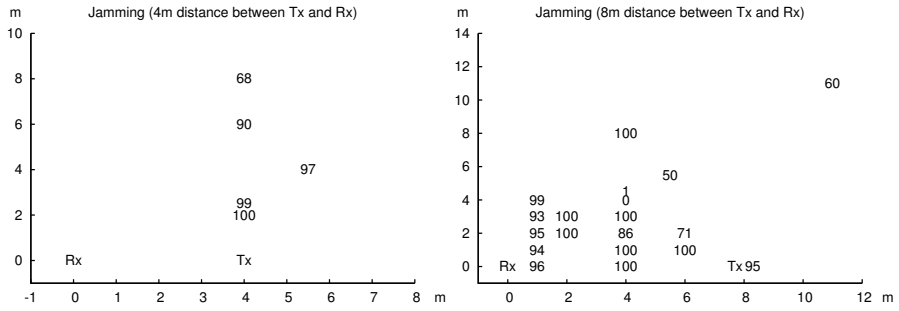


Fig. 4. Success rate of jamming depending on the position of the attacking mote. The Tx and Rx labels are the transmitting and receiving motes. The numbers 0–100 in the graphs denote the percentage of packets that were jammed in particular configurations.

[12] defences against jamming attacks and they propose three basic approaches: changing SFD (start of frame delimiter), shortening frames, and channel hopping. We believe that although the defences may increase complexity of attacks, the efficacy of these three defences varies. The defences also influence reliability of the network communication and incur an increase in the power consumption of the nodes.

Unpredictable SFD Randomising the start-of-frame (SFD) delimiter seems to be a very promising approach as it makes it very hard for the attacker to detect transmitted frames. Unfortunately, available radio chips allow definition of SFD in such a way that SFD is of zero length or its value is *0x00*. This is the value of the frame preamble preceding SFD that can not be changed. The attacker is thus able to eavesdrop all frames regardless on the value of SFD. Changes in SFD value also imply non-compliance with 802.15.4 standard.

Use of short frames It assumes that the shorter the frame the more often the attacker has to listen to detect transmissions. The authors achieved this goal by shortening the preamble as much as possible, and with a fragmentation of frames. The former allowed them to decrease the mandatory data overhead to six bytes (four bytes for PHY header and two bytes for frame check sequence – FCS)[1]. They omitted fcf and dsn fields. Particularly missing fcf, however, would make it very cumbersome to process frames – especially discern data, ACK, beacon, and other types of frames.

There is another serious problem related to the use of shorter preambles – reliability of transmissions. We have experienced problems with quality of the signal even in relatively friendly outdoor environments. Any manipulation of frame formats that decreases the length of the frame headers will influence reliability.

[1] We believe that this overhead would be higher as each PHY frame needs a preamble, SFD, frame length, frame control field, and data sequence number. This would add another three bytes.

Channel hopping It was suggested as a very powerful defence when combined with the frame fragmentation. It will increase the cost of the hardware as more radios must be used in parallel – there are, however, only sixteen channels available, a fact that limits the increase of the cost for attackers.

A serious problem here may be time synchronisation in larger networks. Neither it is clear whether fragments of frames would be delivered in the correct order. A mechanism re-assembling frames (messages) from fragments sent by different motes and belonging to different frames would be needed.

Authors conclude that the probability of frame delivery went down by 20 % with very small transmission distances and just two motes – avoiding the just mentioned aspects.

It seems that jamming is still a problem worth further research. Attacks, as well as defences, may be strengthened and it is not clear whether higher robustness of networks against jamming attacks must necessarily incur higher energy consumption. Some of the defences could be also moved to higher layers of the protocol stack.

You can see that although wireless communication has been with us for a long time, particular technology (frame formats, numbers of available channels, communication speed, and so on) introduces new possibilities for powerful low cost attacks.

7 Stability of network topology

Formation of the network topology is quite important for potential attacks on a network. It is hard to imagine an attacker present during the network deployment, but it is much more likely that the attacker will cause fragmentation or complete disconnection of a network by jamming with the goal to initiate re-establishment of the network connections at their chosen time allowing for active attacks.

7.1 Oscillations

We can demonstrate volatility of the topology even for a very small network (composed of motes on an office desk). We have repeatedly analysed traffic information of a small network of three motes (with IDs 2, 3, and 4) and a gateway (ID 0). We have received similar results when analysing the network installed in an anchorage room of the Humber bridge in the Northern England (see Fig. 6).

Remarkable is also the fact that the quality of links, calculated with a rather sophisticated algorithm by every node in the network, remained very high.

7.2 Traffic analysis

A commercial variant of the MICAz software called XMesh changed addressing of frames. The original version used broadcasting while motes with XMesh address

Subject	Parents
Mote 2	0 for a short while, then repeatedly 3 followed by 0 for briefs
Mote 3	0 and then repeatedly 4 followed by 0 for shorter intervals
Mote 4	0 is assigned as its parent and it remains so

Fig. 5. Topology of a simple network.

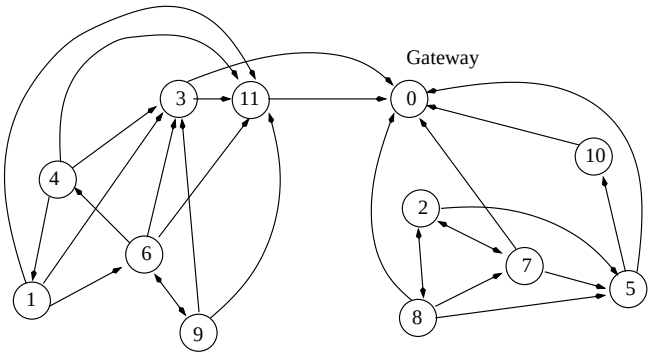


Fig. 6. Graph of all routes appearing in the network deployed on the Humber bridge.

packets to their actual parental motes. Headers with the address cannot be encrypted as they are processed on a very low level of the protocol stack. Use of cryptography would require significant changes in the software and increase processing time and delays required for confirmations of frames delivery.

The attacker can also guess numbers of neighbours from the length of routing packets. Assuming that the attacker is able to jam certain messages, she can easily find the second best neighbour. XMesh will address the second best mote as a parent after six unsuccessful retransmissions of a frame.

It is not sufficient to assume that it is very unlikely for an attacker to be present when a network is being established. Once we start using decentralised, self-forming networks, we allow attackers to bring the networks into a "network state" of their choosing. They can analyse networks and search for the most vulnerable connections even when the communication is encrypted.

8 Attacks on routing

Indeed, network routing seems to be the most vulnerable part of distributed network infrastructures. There are two main reasons for this. The implementations may be vulnerable to malicious attacks, and routing is a distributed algorithm, difficult to control from one point – the gateway, for example, would not able to detect irregularities in the network topology happening only one hop away.

TinyOS and XMesh use sophisticated algorithms built on the number of undelivered messages to compute quality of communication links and to select

the best route to the gateway. Metrics for each direction of communication are treated separately and combined only when a new routing mote is being selected. Messages contain counters allowing for computation of lost messages.

8.1 Forced Selection of Parents

Motes can dynamically change their parental motes according to the numbers of undelivered messages. This feature can be again easily exploited for attacks. One does not even have to jam the communication, just injecting fake messages or replaying old messages with a link quality information would significantly change "quality" of links and the unjammed mote will be selected as a parent.

The parent is always selected according to the link cost computed from the separate numbers of frames lost in each direction. One half of the input information – the number of frames missed by recipients – can be directly forged when transmitted back to the originating mote. The attacker can either lower this estimate, causing the current parental node to be replaced, or improve the estimate for a mote she wants to be selected.

Motes without a route to the gateway are particularly easy to attack and injection of just one message is usually sufficient for the task. Attacks on an already established network are more difficult but there are still two main approaches. The first approach is to jam communication for sufficient amount of time and attack the then disconnected network. The second approach is to lower link quality estimates for all the neighbours except the one we want to become the parent. The latter can be realised by sending spoofed messages to selected motes or by careful jamming of several messages.

It is relatively easy to use selective jamming to change a network topology according to the attacker's objective. It is also notable that this sort of attacks on wireless network is very hard to spot and react upon due to distributed manner of the routing protocol.

8.2 Routing Loops

If the attacker forced a network to create a routing loop, the result would be an enormous increase of the number of messages sent by motes in the loop. What happens is that each message received by any mote in the loop will be forwarded in the loop until it is dropped by one of the motes because its internal buffer of received messages is full or when the message is not acknowledged by any of addressees at some point.

The attack is triggered by injecting a series of messages – one for each mote that is targeted and whose routing table is to be changed and this number does not depend on the length of the resulting loop (see Fig. 7 for simplified attacking code we used with an extended version of Scapy tool).

We have measured number of messages passed over in a loop of three motes at around 40 within 0.8 second. It makes it 16 forwarded messages per mote per second. The level of radio utilisation is however derived from 40 because each

```
mm=ZigBee()/TOSz(type=0xFA,addr=2)/TOS_MH(src=3,orig=3,seqno=355,
        hops=0x00)/TOS_Route(parent=4,cost=0,nbrs=[TOS_RNbr(ID=3),
        TOS_RNbr(ID=4),TOS_RNbr(ID=2)])
nn=ZigBee()/TOSz(type=0xFA,addr=3)/TOS_MH(src=4,orig=4,...
oo=ZigBee()/TOSz(type=0xFA,addr=4)/TOS_MH(src=2,orig=2,...
mm[ZigBee].length=len(mm[TOS_MH])
nn[ZigBee].length=len(nn[TOS_MH])
oo[ZigBee].length=len(oo[TOS_MH])
...
sends(mm);sends(nn);sends(oo)
```

Fig. 7. Python attacking code targeting motes with IDs 2, 3, and 4. It creates complete messages for all three motes and injects them to the network.

mote also listens to all the messages in its proximity. It gives radio utilisation of at least 10% in this instance – ignoring waiting time and transmission for ACK frames. The long term average frequency was just below 30 messages per second.

Once established loop usually holds for a relatively long time. This is due to the fact that a loop eventually increases only the number of hops from the gateway, but this number is not used for routing – link quality computations. This was the case in our experiments when the forwarding was occasionally interrupted only by network management ("route update") messages. Frequency of these messages is in real deployments usually very low.

Implications of this attack on the network lifetime are fundamental and the network would die within tens of hours from complete battery exhaustion. The power requirements for the attacker are, on the other side, very modest.

9 Attacks on cryptographic boundary

When we reimplemented TinySec, a cryptographic library for MICA2 motes, and started using it, we realised several issues arising from optimisation of cryptographic mechanisms for motes with strong power consumption limitations. We mention only one issue to extend the range of attacks that can be launched against sensor networks.

TinySec encryption and integrity protection is really used only on wireless transmissions. Data that left motes via their RS-232 interface is always decrypted.

This property is very useful for system integration. One can decide to switch the TinySec encryption on or off at any time and the gateway will not see any difference – there is no dependency on the back-end part of the wireless system.

On the other hand, the property introduces a new opportunity for an attacker with physical access to some of the motes and ability to connect to their serial (RS232) interface. The attacker can use a legitimate mote to inject arbitrary messages – the mote functions as a cryptographic oracle encrypting and decrypting over-the-air traffic as needed.

The messages sent to the RS-232 interface are by any mote automatically encrypted and transmitted via the motes wireless interface. From the communication point of view motes function as universal transceivers and all messages delivered to a mote are re-transmitted.

10 Conclusions

We have shown how wireless communication technologies change assumptions on which the current security models are based. As one can never make a system perfectly secure, system decisions are based on security risk and threat analysis. Introduction of wireless communication systems not only introduces new threats but also changes risks of the existing ones. As such, communication systems should be subject of new security analysis and possibly redesigned. However, this happens very rarely.

Wireless sensor networks, as any wireless technology, reintroduce many security threats that have been deemed solved or required a very strong attacker. The technology developments squash prices of devices allowing certain attacks to such an extent that even people driven by pure curiosity in a technology can afford them.

Wireless sensor networks also introduce strong decentralisation of many automatic processes that have been in hands of network or system administrators. This shift significantly changes attack vectors and again enables potential adversaries with very low budget, and limited non-technical skills, to attack systems with remote technology-based approaches.

These background changes form the biggest challenge for security. It is very easy to forget why a certain attack was not seen as important. It is very hard to re-think security assumptions when these reasons disappear because of a new way of using products or systems, new technologies, tools, price cuts.

The last aspect really worth noticing is how security is dealt with in the development of wireless sensor networks. The take off of sensor networks is very slow and one would expect there is enough space for designing proper security measures. However, our discussions with Xbow, probably the main player in the area, showed that security is not really an interesting issue until the technology starts being deployed commercially.

References

1. Needham, R., Schroeder, M.: Using encryption for authentication in large networks of computers. Communications of the ACM **21**(12) (1978) 993–999
2. Biryukov, A., Shamir, A., Wagner, D.: Real time cryptanalysis of a5/1 on a pc. In: FSE: Fast Software Encryption, Springer-Verlag (2000) 1–18
3. Barkan, E., Biham, E., Keller, N.: Instant ciphertext-only cryptanalysis of gsm encrypted communication. In: CRYPTO. Number 2729 in LNCS, Springer-Verlag (2003) 600–616
4. Hancke, G.P.: Practical attacks on proximity identification systems (short paper). In: IEEE Symposium on Security and Privacy, IEEE (2006) 328–333

5. Hancke, G.P., Kuhn, M.G.: Attacks on time-of-flight distance bounding channels. In: WISEC. (2008) 194–202
6. Cheung, H.: How to: Building a bluesniper rifle. http://www.tomsguide.com/us/how-to-bluesniper-pt1,review-408.html (2005)
7. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. CCS'02, Washington, DC, USA. (2002) 41–47
8. Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. SP'03, Washington, DC, USA (2003) 197–214
9. Pietro, R.D., Mancini, L.V., Mei, A.: Random key-assignment for secure wireless sensor networks. 1st ACM Workshop Security of Ad Hoc and Sensor Networks Fairfax, Virginia (2003) 62–71
10. Crossbow: Stargate developer's guide (2004)
11. Karlof, C., Sastry, N., Wagner, D.: Tinysec: A link layer security architecture for wireless sensor networks. In: Proc. 2nd SenSys. (2004) 162–175
12. Wood, A.D., Stankovic, J.A., Zhou, G.: Deejam: Defeating energy-efficient jamming in ieee 802.15.4-based wireless networks. In: Proc. 4th IEEE SECON, IEEE (2007) 60–69