[GOR05] Gorecki, C.; Behrens, C.; Westphal, D.; Zheng, C.; Jedermann, R.; Lang, W.; Laur, R.: TAP-SNS - A test platform for secure communication in wireless sensor networks for logistic applications. In: 12th International Sensor Conference. Sensor 2005, AMA Services, Hannover, 2005, pp. 335-340

TAP-SNS – A Test Platform for Secure Communication in Wireless Sensor Networks for Logistic Applications

Gorecki[1], Christian.; Behrens[1], C.; Zheng[1], C.; Westphal[1], D.; Jedermann[2], R.; Lang[2], W. und Laur[1], R.
[1]Institute for Electromagnetic Theory and Microelectronics (ITEM)
[2]Institute for Microsensors, -Actuators and -Systems (IMSAS)
University of Bremen, Department of Physics and Electrical Engineering
Otto-Hahn-Allee, NW1
D-28359 Bremen, Germany

Abstract

In this paper we present a new methodology for the integration of security provisions into protocols for wireless sensor networks. There is an ever growing demand for security in communications in general. As wireless sensor networks (WSN) face real-world applications the authenticity of the retrieved data has to be guaranteed. To facilitate integration of security provisions into WSNs the platform TAP-SNS (Testing Algorithms and Protocols for Sensor Network Security) enables the co-design of secure communication protocols between an x86-architecture and an embedded sensor node platform. Due to the co-design feature of the proposed platform, the use of freely available ANSI C implementations of cryptographic algorithms is made possible. As the system is OSI-model-based, it offers the exchange of whole layers (e.g. serial comport PHY to RF-Link PHY) without the need to adapt the software for the upper layers. The usage of this platform is demonstrated by means of the design of an authenticated broadcast protocol for an application of WSNs in logistics.

Introduction

Recent developments in low-power wireless communication devices and microcontrollers as well as energy-aware standards for data transmission (e.g. IEEE 802.15.4[1])as a step towards the vision of ambient intelligence. Extremely small sensor nodes, integrating sensing, data processing and communication capabilities, are able to form wireless sensor networks. They can be used for monitoring events in physical environments (e.g. fire, leaking of toxic gases) or the state of the goods during the transportation process[2] while providing a long system lifetime. One of the challenges to the design of these networks is to provide reliable, fault tolerant and secure data transmission with low latency times. The versatile requirements for data acquisition can be query-based, event-driven or periodic, depending on the uncounted possible sensor network application scenarios.

A large amount of work has been spent on the development of communication protocols for WSNs, to meet the demands of these scenarios. Aspects like power, computational and memory limitations as well as ad-hoc routing capabilities have to be taken into account[3]. Communication security is one of the key issues for WSN. But most of these proposals have weaknesses as they only address isolated aspects of security like key distribution schemes or authenticated broadcasts[4]. A lot of them assume static network topologies and need extensive pre-configuration of nodes (e.g.[5]). Only very few proposals address more than one or two of these issues. So the development and implementation of secure, ad-hoc capable routing and communication protocols still remains very complex. In most cases, extensive testing and a lot of different revisions of microcontroller software is necessary to create workable solutions. The design of secure communication protocols remains difficult without extended debugging features and the possibility to easily create a graphical user interface for optimization purposes and for the identification of timing problems or encryption errors.

In this paper we present the novel test platform TAP-SNS as an implementation aid for security provisions in wireless sensor networks[6]. It enables co-design of secure communication protocol software between an x86-architecture and the embedded device. TAP-SNS helps to identify the bottlenecks of the used algorithms (e.g. memory usage, processing time etc.) *before* the implementation on the embedded hardware and provides advanced debugging features of the x86-architecture for the development and test of security protocol implementations.

The workflow using TAP-SNS (Figure 1) is altered by the incorporation of the x86-platform. Various cryptographic primitives can be prototyped and tuned for proper cooperation. The functionality (e.g. Timing, Handshake) can be verified using TAP-SNS. After the protocol has been tested successfully, the developer should be provided with an API consisting of functions for secure communication, the so-called security sublayer. Based on this sublayer a routing scheme has to be implemented. After additional testing the communication solution can be ported to the desired microcontroller platform.

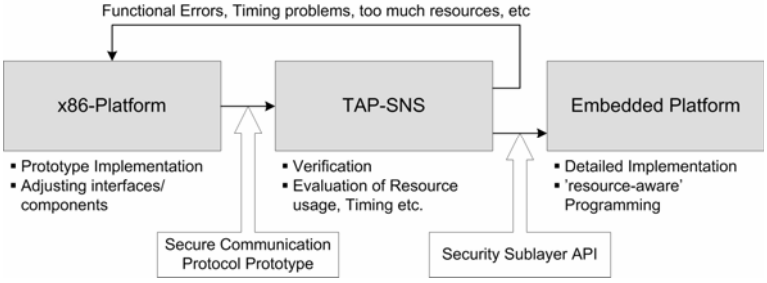


Figure 1: Workflow for the usage of TAP-SNS

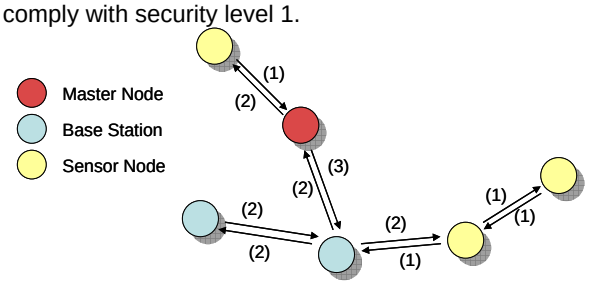
Another immediate advantage of TAP-SNS is the use of standard off-the-shelf ANSI C implementations of various cryptographic algorithms (e.g.[7]) for rapid-prototyping security solutions. This will reduce the total effort for implementing secure communication protocols in WSNs.

Proposed WSN platform

The wireless sensor network platform is based on a three-tier-hierarchy. On top of the system is a *master node*. It acts as central data collection and processing point and a gateway to other networks (via e.g. WLAN, UMTS). Its hardware is comparable to a PDA (e.g. ARM7). The middle tier element is called *base station*. The main task of a base station is to fulfill its role as a cluster head in a cluster-based routing scheme (e.g.[8]). Data aggregation is implied in this role. The hardware for this network node comprises of a Texas Instruments MSP430F1611 16-bit microcontroller[9] in conjunction with a Chipcon CC2420 IEEE 802.15.4 RF transceiver[10]. The lowest hierarchy elements are sensor nodes. The sensor node's main objective is the collection of environmental data. The sensors can be fitted to this platform in an application- and mission-specific manner (e.g. temperature, humidity, gases, radiation) and are connected using an interface µC and a SPI-Bus to the main CPU. The hardware platform of these nodes can be the same as for the base station (for a dual-purpose base station/sensor node functionality) or can feature a smaller microcontroller with reduced resources and extended lifetime.

Secure Routing with a layered addressing scheme

Three different levels of security can be identified in this system. They correspond to the impact, that an attacker will be able to exert, if he captures a node on a certain level of the hierarchical structure of this sensor network (Figure 2). According to the nodes' tasks, their hardware capabilities allow the execution of different security functions. As outlined above sensor nodes don't have the necessary capabilities to carry out extensive cryptographic functions like public key cryptography. If an attacker compromises a single sensor node, his impact on the overall network security is marginal, so that the security requirements for the communication link between sensor nodes as well as from a sensor node to a base station comply with security level 1.



	Master Node		*Base Station*		*Sensor Node*	
	Up-Link	*Down-Link*	*Up-Link*	*Down-Link*	*Up-Link*	*Down-Link*
Master Node	(x)	(x)	2	3	1	1
Base Station	2	3	2	2	1	2
Sensor Node	1	2	1	2	1	1

Figure 2: Security levels for a hierarchical sensor network topology

Figure 3 shows the communication structure between sensor nodes and a base station inside a cluster. Sensor nodes which are performing same physical measurements (e.g. temperature or humidity) are organized in chains. The idea for organizing nodes in chains is known from the PEGASIS proposal (Power-Efficient Gathering in Sensor Information Systems)[11]. Unlike PEGASIS, our structure is not based on the ability of nodes to have knowledge of their actual residence and the nodes also don't need to have a base station within a 1 hop distance. The setup of chains is performed inside of each cluster and based on the sensor type which is connected to the sensor node. The sensor type is accessible and known by the system.

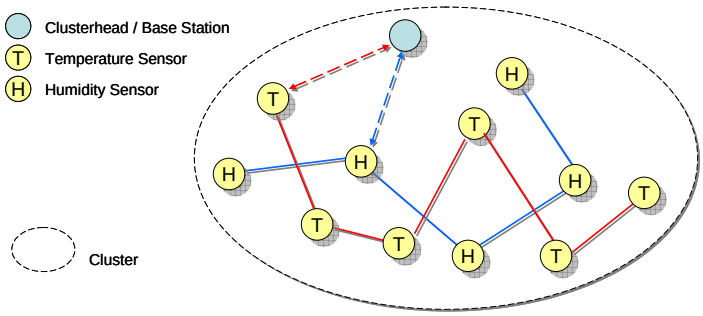


Figure 3: Chain structure inside a cluster

One benefit of our proposal is the small size of the routing table inside each sensor node. From network security's point of view, the routing table is organized like an access control list (ACL, e.g.[1]) which not only has to deal with addresses but also with keys, counter values and initialization vectors (IVs). So the number of entries should be as

small as possible. If the size of a cluster is e.g. limited to 25 nodes, which are divided into two or more chains, the resulting maximum size of the routing table is limited to 25 entries (if all sensor nodes are from the same type). Time is divided into intervals (*rounds*) which are related to the IEEE 802.15.4 super frames provided by the cluster head. One node per interval reports the measurement results of his chain to the cluster head. This approach keeps energy requirements as low as possible and allows qualified majority decisions.
A sensor node which is not the reporting node of an actual round only has to communicate with his immediate neighbor, which is chosen based on sensor type and the RSSI signal (Received Signal Strength Indication), which is provided by the 802.15.4 transceiver chip.
In our approach the mode of operation in each cluster can be switched between *on demand* (OD), *interval based continuous* (IBC) and *threshold driven* (TD) measurements. In TD-mode, hard and soft threshold values are advertised to each chain by the cluster head. The sensor nodes only need to report a violation of e.g. the allowed maximum value for humidity. The use of threshold values is known from the TEEN (Threshold sensitive Energy Efficient Network Protocol) proposal[12]. The nodes only need to receive the super frame beacons to keep their key scheduling synchronized and are allowed to switch off their radio for the rest of time.
Cluster heads / base stations are responsible for data aggregation and compression within their clusters. They also provide the key scheduling and management in their their cluster. The link from a cluster head to a chains node or to all nodes within his cluster (broadcast message) implies higher security risks compared to links between sensor nodes. If an attacker captures such a link, he is able to control a chain of nodes or the whole cluster, so the security requirements for such links comply with security level 2. Based on their larger hardware capabilities and energy resources, cluster heads are able to perform not exclusively the symmetric cryptographic algorithms for secure communication with sensor nodes, but also asymmetric algorithms, which are used to establish secure channels between cluster heads or for the communication with the master node. Therefore the routing layer makes use of different cryptographic primitives, offered by the API of the security sublayer. The modular layout of this layer is denoted in Figure 4 to give a simplified overview. The crypto modules can easily be combined according to the capabilities of the nodes type.
The master node represents the 802.15.4 PAN-Coordinator and is responsible for the whole network. He distributes tasks to the cluster heads and acts as trusted device during the node registration process. Therefore the link from master node to cluster heads has to be highly secured. This is represented by security level 3.

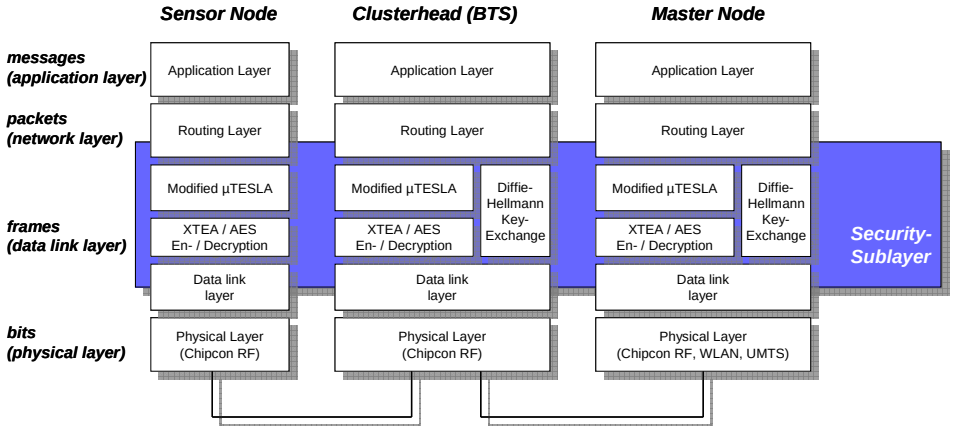


Figure 4: Simplified security sublayer

In the proposed system, there is no need for the master node to address a certain sensor node somewhere in the network which is not directly a member of his own cluster. Due to the fact, that cluster heads report the abilities of their sensor node chains to the master node, a simple attribute-based routing algorithm[13] can be used inside the master node. It is combined with direct accessible links to the cluster heads, corresponding to their entries in master nodes access control list, which is completed with the attributes provided by the cluster heads. In this way we keep the routing effort on each level of our hierarchical topology as small as possible. Estrin et al.[13] proposed data centricity and application specificity design features for wireless sensor networks. By using application specificity, a sensor network can be tailored to its actual task. If no GPS-chips are used to provide geographic information, this can also be realized through the master node's external network connection. The master node can be supplied with localized information (e.g. where a certain cluster head has been installed). The attribute-based secured addressing scheme is often revered to be the most ideal scheme for wireless sensor networks[14]. The presented approach leads to a layered and attribute-based secured addressing scheme, which minimizes the communication effort and the memory requirements for routing tables.

Solution by the example of authenticated broadcast

In our wireless sensor network, cluster heads use authenticated broadcasts during cluster formation and for key distribution purposes. Perrig et al. proposed SPINS (Security protocols for sensor networks)[15] in 2001. SPINS consists of SNEP (Secure Network Encryption Protocol) and µTESLA ("Micro" version of the Timed Efficient Stream Loss-tolerant Authentication Protokolls TESLA[16]) which is used for authenticated broadcasts.The main idea which is adopted for our solution is the use of a single crypto module (a symmetrical block cipher) which is used in different

operation modes for multiple operations. We chose the XTEA cipher[17], which needs less memory than the RC5 Algorithm proposed in[15]. A 128 byte key is used for all of our implementations. The results are given in Table 1.

Memory Requirements of Symmetric Blockciphers in Byte (MSP430F149 / IAR compiler)		
Block Cipher	Code Segment	Data Segment
AES	14524	0
RC5 (12/16)	1766	0
RC2	992	256
XTEA	576	0

Table 1: Memory requirements of different symmetric Block Ciphers [3]

µTESLA requires that the cluster head and the nodes are loosely time synchronized, and each node needs to know a threshold value of the maximally allowed synchronization error. The synchronization is achieved by use of the *beacon-enabled* mode of our IEEE 802.15.4 transceiver chips. The cluster head computes a one-way key chain by repeatedly applying a public known one-way function *F* to a randomly chosen key K_n and starts the disclosure of keys from the last key of the chain (K_0; K_1, K_2, K_3, ...). Due to the one-way property of *F*, every node can compute K_{n-1} from K_n but no one is able to find a key K_{n-1}, so that $F(K_{n-1})$ gives K_n (the valid key of the next time interval). To authenticate a broadcasted packet, the cluster head computes a message authentication code (MAC) on the packet with a key that is secret at that point in time. When the node gets the packet, it can verify that the corresponding MAC key was not yet disclosed by the cluster head. To do so, the sensor node must have knowledge of the time schedule for the key disclosure. If the sensor node is assured that the MAC key is not a previously disclosed one, it stores the packet in a buffer. At the time of key disclosure, the cluster head broadcasts a key disclosure packet to all nodes in its cluster. By use of this key, a sensor node is able to verify the authenticity of the key and also the MAC of the previous received packet in its buffer. Every sensor node needs at least one authentic key of the chain to be able to receive authenticated broadcasts. This key has to be transmitted over a secured channel.
A modified µTESLA implementation is used is used in the proposed system to deliver the cluster key to sensor nodes in an authenticated manner. During the registration process the sensor node generates a so-called one-way ticket which has to be signed by the master node. If access to network is granted by the master node, both, the cluster head and the sensor node compute a one-time key which is used to set up a secure channel via the hardware AES module of the Chipcon transceiver. This authenticated and encrypted link is used to transfer a key of the one-way key chain the sensor node needs to participate in authenticated broadcasts. The main benefit of our approach is that the key that is used to secure this link is never transferred over the network. Figure 5 shows a cut-out from the preliminary design of the message flow between the software modules located within the cluster head / base station during the registration process of a sensor node. The basic µTESLA was furthermore adapted to the hierarchical cluster-based network topology and functionality was added to support forwarding and mobility of sensor nodes. But these topics are beyond the scope of this paper.

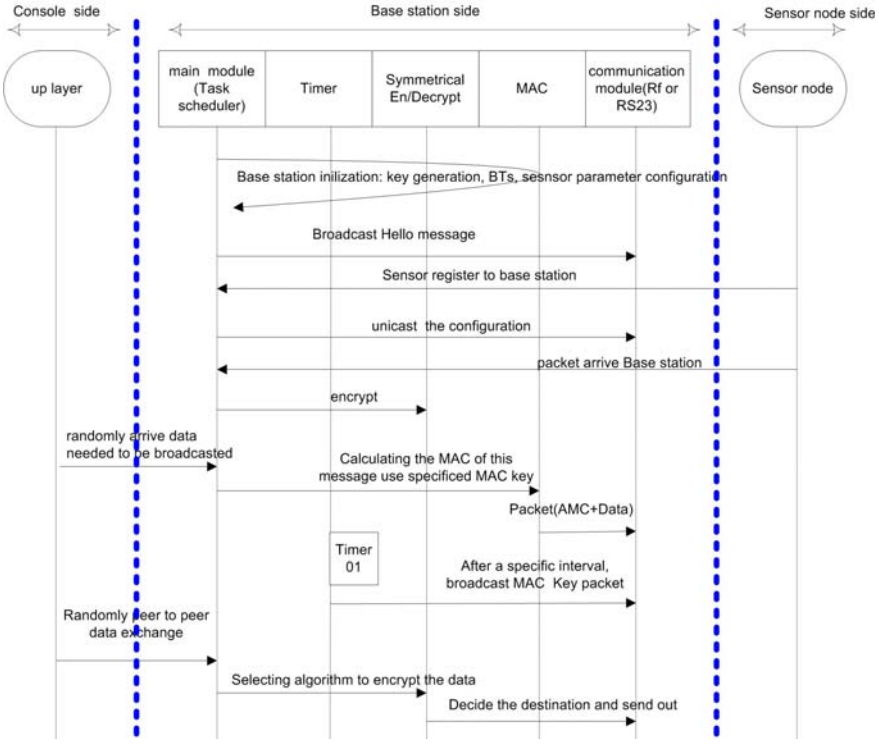


Figure 5: Preliminary design of the message flow between SW-modules during node registration

The idea behind TAP-SNS is to breakdown the test and implementation process to small sized modules which can later be combined to make up the communication software of sensor nodes or cluster heads. The layer-based architecture abstracts from the transport medium and helps to focus on message creation and handling on both sides of the communication link. Packets can easily be added or changed by defining a new message ID or modifying an existing message type. The cryptographic algorithms are compiled to a library which is linked to the application and interfaced via a standardized interface. This allows switching between different ciphers and modes of operation. Due to this feature, the different cipher primitives may be used for data en-/decryption or for MAC-generation/-verification either. Debug-messages can be displayed in the graphical user interface to control proper function of modules. The software-modules are written in ANSI-C to allow easy migration of the software to the embedded system. The interface to these modules is written in C++. In our example this interface is provided by the *µTESLA class*.

Application of TAP-SNS

After the design of the message flow (Figure 5) the needed functions are provided in two elements. The *encriptlib.lib* contains functions for the use of cipher primitives (in our implementation AES, RC2, RC5 and XTEA). The other element is a class (in this case the µTesla class) that contains all functions that are necessary for secure communication by use of the functions provided in the cryptographic library. TAP-SNS is started using the *wsnmanager* (Figure 6)

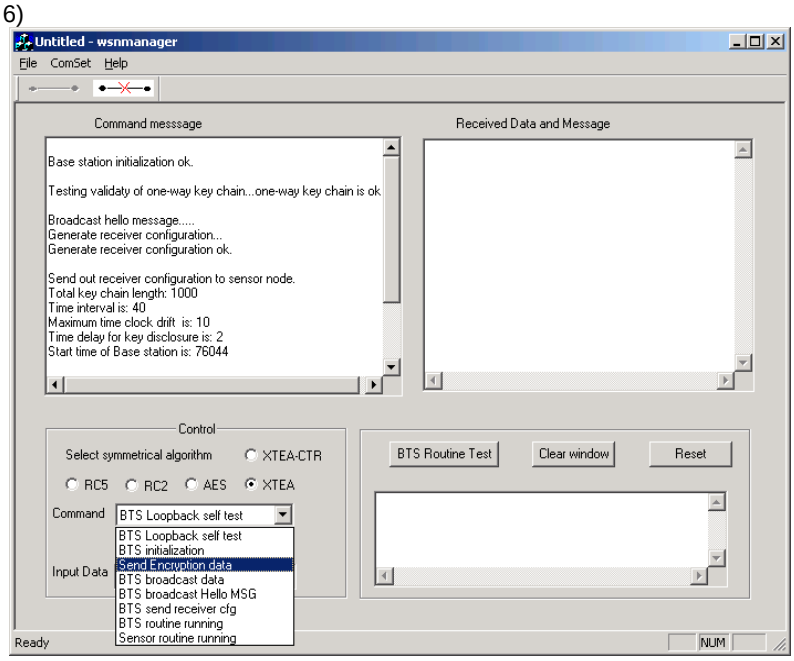


Figure 6: The *wsnmanager* Base Station/Master Node window

Figure 7 depicts the integration of the µTESLA class into the wsnmanager software. In this figure the function view of the commands on sensor node side is shown. These commands enable packet-parsing in the authenticated broadcast procedure.

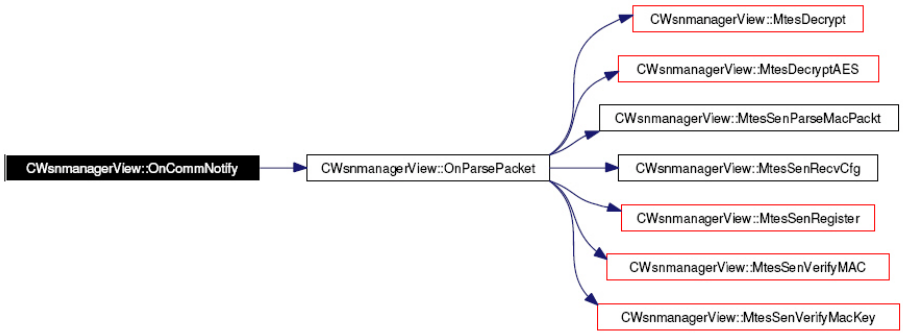


Figure 7: Example for a sensor nodes µTESLA interface via the wsnmanager

The wsnmanger shows a window on each PC. One for the base station/master node and the other window for the sensor node. From the base staion/master node window commands can be issued, that directly correspond to the functions inside the class. The sensor node window reacts to these commands. The generated message flow of the nodes is visualized in both wsnmanger windows. After successful test of the protocol, the *security sublayer* (the foundation of the cryptographic library and the communication class) can be ported to the microcontroller. Due to the OSI-based approach, it is possible to easily exchange the communication media (RF-Link for serial link). The porting process to the microcontroller platform is visualized in Figure 8.

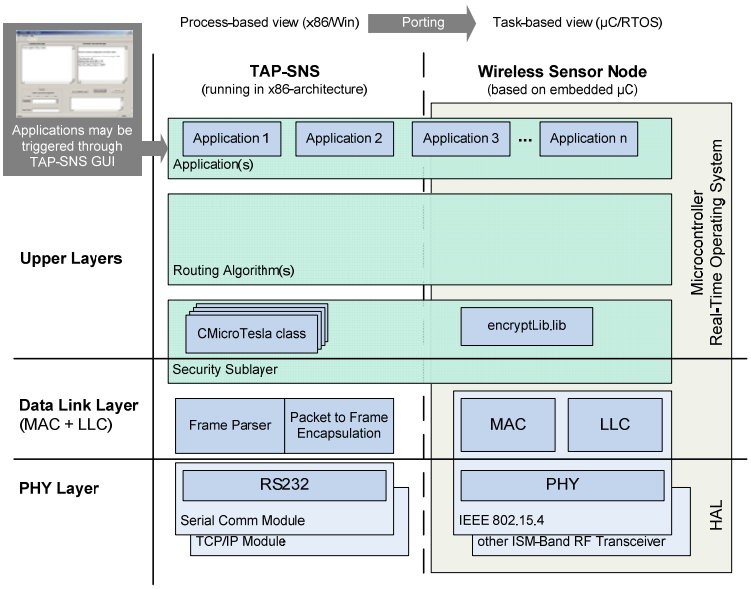


Figure 8: Application of TAP-SNS and porting issues

Conclusions and Future Work

TAP-SNS significantly eases the implementation of security provisions into communication protocols of wireless sensor networks. The next step is the enhancement of TAP-SNS from peer-to-peer-only communication to multi-node and multi-hop topologies to fully represent the hierarchical structure of the proposed WSN system.
But still, the design of communication protocols itself remains a very complex task. There is an absolute need for tools that facilitate the design of communication protocols. A proposed goal would be a protocol design tool that enables a developer to design his own communication protocol inside an integrated development environment . The length of protocol units, routing algorithms, security provisions, PHY, etc. are added via drag-and-drop while the tool provides the source code for the implementation, that can be directly ported to a selected microcontroller. But there is still a lot of work to be done to fullfill this vision.

Literature

[1] Jose A. Gutierrez et.al.: IEEE 802.15.4 "Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensor Networks, IEEE Press, ISBN 0738135577
[2] M. Freitag, et.al.: "Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel und seine Grenzen“, in Industrie Management 20 (2004).
[3] J. Polastre et. al.:“The Mote Revolution: Low Power Wireless Sensor Network Devices”, in Proc. Hot Chips 16: Aug. 2004
[4] B. Duncan, D. Malan: “Low-Power, Secure Routing for the MICA2 Mote”, Harvard University, Technical Report TR-06-04, Mar. 2004
[5] H. Chan et.al.: "Random key predistribution schemes for sensor networks", in IEEE Symposium on Research in Security and Privacy,2003
[6] C. Zheng: “Security in Wireless Sensor Networks for Autonomous Logistics Systems”, Master Thesis, University of Bremen, October 2004
[7] http://www.cs.auckland.ac.nz/~pgut001/ cryptlib/
[8] W. Heinzelman et. al.: "Energy-efficient communication protocol for wireless sensor networks," in Proc. HICSS-2000, Hawaii, Jan. 2000.
[9] “TI MSP430F1611 Device Datasheet”, http://www.ti.com/msp430
[10] “Chipcon SmartRF CC2420 Preliminary Datasheet”, http://www.chipcon.com
[11] S. Lindsey and C. Raghavendra: “PEGASIS: Power-efficient gathering in sensor information systems”, in IEEE Aerospace Conference, Mar. 2002.
[12] A. Manjeshwar and D.P. Agrawal: “TEEN: A routing protocol for enhanced effeciency in wireless sensor networks”, WPIM 2002, IPDPS Workshop, 2002.
[13] D. Estrin et. al.: “Next century challenges: Scalable coordination in sensor networks”, in 5th annual ACM/IEEE International Conference on Mobile Computing and Networking, 1999,
[14] J. Ibriq, I. Mahgoub: “Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges”, Proc. of SPECTS'04, 2004
[15] A. Perrig et. al.: “SPINS: Security protocols for sensor networks”, in Proc. of Mobicom 2001, July 2001
[16] A. Perrig et.al. :“The TESLA Broadcast Authentication Protocol”, Technical Report, UC Berkley, IBM Research, 2002
[17] R. Needham and D. Wheeler: “Tea extensions”, Technical report, Computer Laboratory, University of Cambridge, Oct. 1997