Reliable Broadcast Message Authentication in Wireless Sensor Networks

Taketsugu Yao, Shigeru Fukunaga, and Toshihisa Nakai

Ubiquitous System Laboratories, Corporate Research & Development Center, Oki Electric Industry Co., Ltd., 2-5-7 Honmachi, Chuo-ku, Osaka, Japan
{yao282, fukunaga444, nakai365}@oki.com

Abstract. Due to the low-cost nature of sensor network nodes, we cannot generally assume the availability of a high-performance CPU and tamper-resistant hardware. Firstly, we propose a reliable broadcast message authentication working under the above-mentioned circumstances. The proposed scheme, although based on symmetric cryptographic primitives, is secure against anyone who knew the message authentication key as well as the malicious router nodes in multi-hop networks. The proposed scheme consists of three steps; (i) reliable broadcast of a message, (ii) legitimate acknowledgments from all the nodes in the network, and (iii) disclosure of the message authentication key. Secondly, we propose a way to reduce the amount of the stored information until the disclosure of the key, in which the server transmits the message integrity code of a message before transmitting the message. Finally, we consider the characteristic and the security issues of the proposed schemes.

Keywords: Sensor Networks, Software Update, Message Authentication, One-way Key Chain, Secure Acknowledgment, Symmetric-key.

1 Introduction

Recently, sensor networks are proposed for a wide variety of applications, such as home and building automation systems, industrial plant management systems, and environmental monitoring systems. We suppose that the sensor network system consists of a large number of resource-constrained sensor nodes and the server which manages and controls the system. In this system, the server and all nodes transmit and receive data using wireless multi-hop networks.

Our target is updating software on sensor nodes over wireless networks. As mentioned above, there are a large number of sensor nodes in sensor networks. If we find software bugs on sensor nodes or try to add new functions to them, it takes lots of works to pick up all nodes and update the software in nodes. Updating software on nodes over wireless networks, therefore, is effective in the remote maintenance of the sensor networks, such that the server transmits update data to nodes, and nodes update their software by themselves.

Security is one of the essential issues in updating software on sensor nodes. We suppose that software update should satisfy the following two requirements.

X. Zhou et al. (Eds.): EUC Workshops 2006, LNCS 4097, pp. 271 – 280, 2006.

(i) Each node authenticates received data as one signed by the certified server.
(ii) The server receives acknowledgements that all nodes are sure to receive the correct update data.

Related works are TESLA: [1],[2],[3],etc, and Merkle hash tree: [4],[5],etc. TESLA offers sender authentication at the cost of loose initial time synchronization and slightly delayed authentication. Merkle hash tree is a tree-chaining techniques for signing/verifying multiple packets using a single signing/verification.

In this paper, we propose a message authentication scheme that satisfies the above two requirements in multi-hop tree networks. Our scheme needs no time synchronization between the sender and the receivers. Instead, the key disclosure appeared in TESLA relies on the reception (by the sender) of secure acknowledgements from the receivers. Here, we use a Merkle hash tree-like technique in a secure acknowledgement scheme.

The rest of this paper is organized as follows. In section 2, we explain the constraints of sensor nodes on hardware and the desired message authentication for sensor networks. In section 3, we introduce the message authentication using a one-way key chain construction. In section 4, we propose a reliable broadcast message authentication which is characterized by the secure acknowledgments of broadcast data. In section 5, we consider the characteristics of the proposed schemes and also describe the security issues. Finally, we conclude this paper in section 6.

2 Constraints and Desired Message Authentication for Sensor Networks

Due to the low-cost nature of sensor network nodes, we cannot generally assume the availability of a high-performance CPU and tamper-resistant memory. We desire a

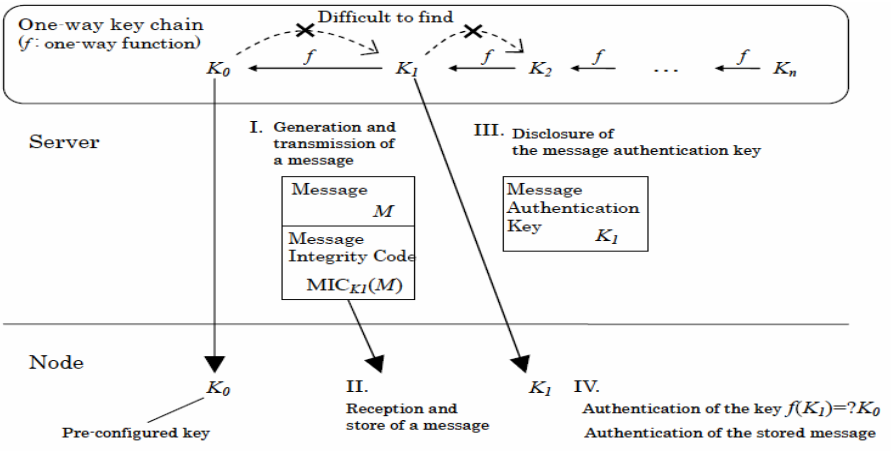

Fig. 1. Message authentication using a one-way key chain

broadcast message authentication for sensor networks, such that nodes can authenticate the messages without secret information which is characterized in asymmetric-key-based schemes, as well as nodes can authenticate the messages with less computational power like symmetric-key-based schemes.

A broadcast message authentication based on symmetric-key cryptographic primitives is generally implemented by generating/verifying the message integrity code (MIC), which is also called the message authentication code (MAC), using the message authentication key shared among the server and all the other nodes.

On the other hand, a broadcast message authentication based on asymmetric-key cryptographic primitives has the advantage that there is no necessity for nodes to conceal the message authentication key (generally called a public-key). There have been studies on evaluating the calculation cost of asymmetric-key cryptographic primitives in an effort to be performed by sensor nodes: [6],[7],[8], etc.

3 Message Authentication Using One-Way Key Chain

In this section, we introduce a message authentication scheme relying on a one-way key chain construction which is the basic idea for TESLA: [1],[2],[3]. This scheme has the advantage that nodes can authenticate the messages without secret information, although it is based on symmetric-key cryptographic primitives.

3.1 Procedures

A one-way key chain is a chain of keys generated through repeatedly applying a one-way function on a random number. A one-way function is one which is easy to compute but difficult to invert. A message authentication scheme using a one-way key chain uses each of keys in a one-way key chain as the message authentication key, which is used for generating/verifying MICs.

In figure 1, we show an example of the procedure of the message authentication scheme using a one-way key chain. In this scheme, the stored key in nodes is for verifying the next key in the one-way key chain, and does not have to be secret. Even if attackers compromise and fraudulently obtain the stored key in nodes, it is difficult, on the principle of a one-way function, for the attackers to find the message authentication key to be used for generating the correct MICs of messages.

3.2 Threats in Multi-hop Communication Environments

The scheme explained in section 3.1 has the drawback that the nodes, which know the key disclosed by the server, may spoof the nodes which have not yet known the key. For example, router nodes in multi-hop networks can impersonate the server by delaying the messages intentionally. A malicious router node does not forward the message transmitted by the server to the next hop nodes and waits for the disclosure of the message authentication key by the server. When the message authentication key is disclosed by the server, the malicious node generates the MIC of malicious messages using the disclosed key, and forwards the malicious messages with its MICs to the next hop nodes. After that, the malicious node forwards the disclosed key to the

next hop nodes. Consequently, the malicious node can deceive all descendant nodes into accepting the malicious messages as the authenticated ones.

4 Reliable Broadcast Message Authentication

In this section, we propose a reliable broadcast message authentication which is tolerant to the server spoofing attacks even in multi-hop networks. The server can confirm that all nodes are sure to receive the correct data because our scheme makes use of secure acknowledgments (ACKs).

4.1 Outline of the Proposed Scheme

As mentioned in section 3.2, there is a risk that the attackers may know the disclosed key earlier than victim nodes in multi-hop networks. Therefore, it is desirable that the message authentication key becomes invalid once the server discloses it.

The proposed scheme has the following two features:

(i) The server discloses the message authentication key after the server has received the correct ACKs from all the other nodes.
(ii) The system specifies the number of the messages to be authenticated per each of keys in a one-way key chain, and the server and all the other nodes synchronize the number of authentications.

The proposed scheme works well and the attacker cannot deceive the victim nodes as long as the scheme prevents the attacker from forging the acknowledgement messages.

In the next section, we introduce a secure and efficient acknowledgment scheme for multi-hop tree structured networks as an example to realize the system mentioned above.

4.2 Secure Acknowledgements Adopted in Multi-hop Tree Networks

In this section, we introduce a secure and efficient acknowledgment scheme of the broadcast messages transmitted by the server for multi-hop tree networks which is typical structure of sensor networks.

There are the following problems in the acknowledgments of the broadcast messages transmitted by the server in multi-hop environments.

(i) Replying the ACKs causes a large transmission overhead, especially under the circumstances in which there are a large number of nodes in sensor networks.
(ii) The ACKs lack authenticity. (For example, it is possible for malicious router nodes to forge ACKs.)

Ariadne[9] is a secure routing protocol. We adopt the concept of a per-hop hashing introduced in Ariadne to the generation of ACKs for multi-hop tree structured networks. We model the multi-hop tree network as a Merkle tree, and generate ACKs using a Merkel hash tree-like techniques. In this scheme, we assume the server and all the other nodes previously share the routing information and pair-wise distinct keys, where each of these keys is shared between the server and a unique node of the

network. We show the basic concept of our acknowledgment scheme in figure 2, where M is a broadcast message from the server, K_A, K_B, K_C, K_D, and K_E is pair-wise distinct key shared between the server and the node A, B, C, D, and E, respectively, and h_A, h_B, h_C, h_D, and h_E is secure ACKs generated by the node A, B, C, D, and E, respectively.

Figure 2 (a) illustrates the ACK generation in the proposed scheme. Equation (1) shows a secure ACK h_X generated by a node X.

$$h_X = \mathrm{H}(\mathrm{MIC}_{K_X}(M) \{ \| [Secure\ ACKs\ from\ the\ children\ nodes] \| \ldots \}) \quad (1)$$

where $\mathrm{MIC}_{K_X}(\cdot)$ represents a MIC generation algorithm using a pair-wise distinct key K_X, H represents a hash function, $\|$ represents a bit concatenation, and "{ $\|$ [*Secure ACKs from the children nodes*] $\|$ …}" represents secure ACKs generated by the children nodes of a node X when a node X is a router node.

Figure 2 (b) illustrates the verification procedure of the secure ACKs by the server. The server has the node control table as shown in figure 2 (b) to check pair-wise distinct keys and the routing information of all nodes in the network. The server can compute secure ACKs using its broadcast message M, pair-wise distinct keys, and the routing information in the node control table as nodes have generated. The server verifies whether or not its own computing information h'_A equals to the secure ACK h_A replied from nodes, and if it equals, the server acknowledges that the broadcast message M is certainly reached to all of the nodes.

The secure acknowledgment scheme as mentioned above has the advantage that the server can confirm not only that all nodes receive the correct message, but also that there is no change in the network structure which has been known by the server.

This scheme can suppress the transmission overhead compared with the case that each node replies an ACK to the server. Moreover, a secure ACK is generated using one or more pair-wise distinct keys shared by between the server and a unique node. So it is difficult for the attackers to forge any ACKs forwarded by any nodes without knowing all pair-wise distinct keys which is involved in the generation of the ACKs.

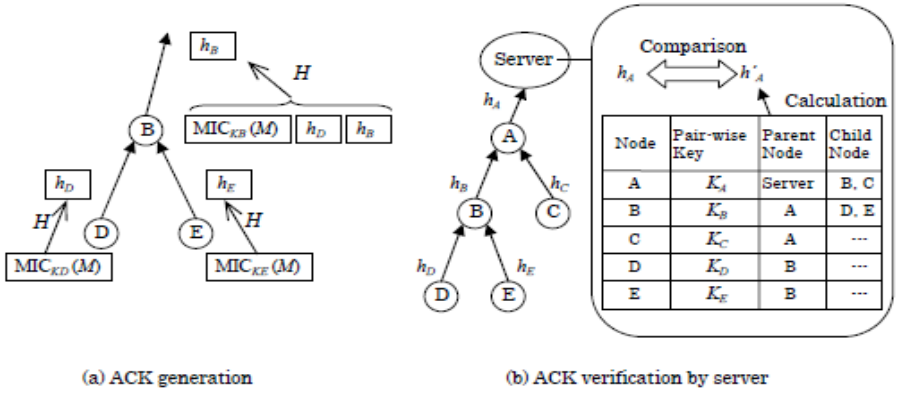

Fig. 2. Secure and efficient acknowledgments in multi-hop tree structured networks based on the concept of a per-hop hashing

On the other hand, there are drawbacks in this scheme. It allows any corrupted node to make the whole authentication scheme collapse by injecting a single incorrect piece of data during transmission of ACKs to the server. Despite the server know all the keys, the server cannot invert the hash function. Therefore the server is not able to detect which nodes have been corrupted in the network. A way of solving this problem may be that router nodes aggregate the ACKs using a reversible algorithm such as an exclusive-or on behalf of a hash function.

4.3 Implementation

Figure 3 illustrates the implementation of the proposed message authentication scheme, which combines the message authentication scheme using a one-way key chain as mentioned in section 3.1 and the secure acknowledgment scheme as mentioned in section 4.2. If the system specifies that several messages can be authenticated per each of keys in a one-way key chain, the server should disclose the message authentication key after verifying the acknowledgments of all messages.

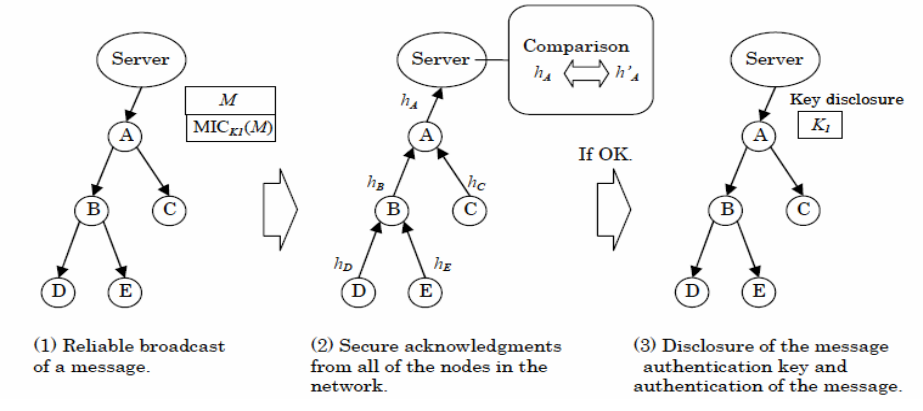

Fig. 3. Reliable broadcast message authentication

4.4 Store-Reduced Version of Our Scheme

The proposed scheme, as mentioned above, has the issue that the nodes must store the received messages until the server discloses the message authentication key because the nodes cannot authenticate the messages without the key. This could be critical to memory-constrained nodes. In this section, we propose a way to reduce the amount of the stored information until the disclosure of the key. In this store-reduced version of the proposed scheme, the server transmits the MIC of a message before transmitting the message. After verifying the correctness of ACKs of transmitted data from all the nodes, the server transmits the message authentication key and the message corresponding to the already transmitted MIC. In this case, the nodes can authenticate the message immediately when it is received, and the amount of the stored information in nodes until the key disclosure is only the MIC, whose data size is

generally assumed to be smaller than data size of the message. Thus we can reduce data size to be stored in the nodes until the nodes know the message authentication key.

The store-reduced version of the proposed scheme does not satisfy the following requirement at the time of the verification of the correctness of the received ACKs of the MICs: the server receives acknowledgments that all nodes are sure to receive the correct messages. Indeed, the above requirement can be satisfied by the server transmitting a combination of the following data: the message authentication key, the message, and the MIC which is generated by the next message to be going to be transmitted. The nodes reply the ACKs of the combination data. If the server verifies the correctness of the received ACKs, the server acknowledges that all nodes were sure to receive the correct messages as well as the correct MICs.

Table 1. Comparisons among the four message authentication schemes based on symmetric-key cryptographic primitives about both the tolerance to the server spoofing attacks and the receipt of correct data by nodes, where is (a) message authentication using the key previously shared by among the server and all the other nodes, (b) message authentication using a one-way key chain, (c) TESLA, and (d) the proposed scheme, respectively

	(a)	(b)	(c)	(d)
Tolerance to the server spoofing attacks	Poor	Good	Best	Best
Receipt of correct data by the nodes	Not confirmed	Not confirmed	Not confirmed	Confirmed

5 Security Considerations

In this section, we consider the characteristic of the proposed scheme and also describe the security about the tolerance to the server spoofing attacks of the proposed ones.

Firstly, we compare the proposed scheme about the characteristic with other three schemes based on symmetric-key cryptographic primitives: (a) message authentication using a key previously shared by among the server and all the other nodes as mentioned in section 2, (b) message authentication using a one-way key chain as mentioned in section 3, and (c) TESLA: [1],[2],[3], respectively, as shown in table 1. The proposed scheme is tolerant to the server spoofing attacks even in multi-hop environments on the assumption that it is difficult for the attackers to forge the ACKs of data transmitted by the server. In addition, our scheme needs no time synchronization between the server and all the other nodes to disclose the message authentication key used to generate/verify the MICs. Instead, the server verifies that all nodes reliably receive the correct messages, because key disclosure of our scheme relies on the reception (by the sender) of secure ACKs from the receivers.

Secondly, we consider about the tolerance to the server spoofing attacks of both the proposed scheme and the store-reduced version of the proposed scheme.

In the proposed scheme, it is guaranteed that the valid messages have reached to all of the nodes at the time of the key disclosure, because the server transmits the

message with its MIC and discloses the message authentication key after the legitimate acknowledgments of both the message and the MIC. So even if the malicious router nodes put the malicious messages into the networks, the nodes can dispose of them as follows:

(i) Against the malicious messages which are put before the key disclosure. We assume that the malicious router nodes put the malicious messages into the network before the key disclosure. In this case, the malicious router nodes cannot generate the correct MICs corresponding to the malicious messages because the message authentication key has not yet been disclosed. The nodes can dispose of the malicious messages depending on the verification results of the MICs.

(ii) Against the malicious messages which are put after the key disclosure. We assume that the malicious router nodes put the malicious messages into the network after the key disclosure. In this case, the malicious router nodes can generate the correct MICs corresponding to the malicious messages because the message authentication key has already been disclosed. However, it is guaranteed that the valid messages with its MICs have reached to all of the nodes at the time of the key disclosure. The nodes can dispose of the malicious messages depending on the received order because the malicious messages are received after all nodes have received the valid messages with its MICs as shown in figure 4. Here, the system should specify the number of the authenticated messages per each of keys in a one-way key chain.

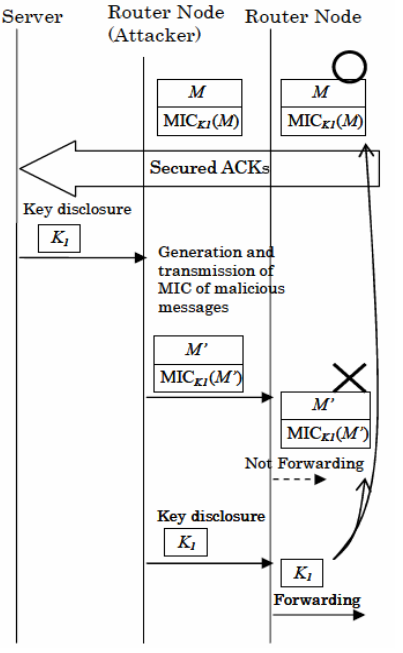

Fig. 4. Example of nodes disposing of malicious messages in the proposed scheme

Table 2. Comparisons of the proposed schemes about both stock data in nodes until the key disclosure, and the receipt of MICs more than the specified number per each of the keys in a one –way key chain

	The proposed scheme	The store-reduced version of the proposed scheme
Stocked data in nodes until the key disclosure	Message and its MIC	MIC
Receipt of MICs more than the specified number per each of the keys in a one-way key chain	Acceptable	Not acceptable

In contrast, in the store-reduced version of the proposed scheme, the server transmits only the MIC at first, and discloses both the key and the message after verifying the correctness of the received ACK of the MIC. Therefore, it is not guaranteed that the valid messages have reached to all of the nodes at the time of the key disclosure. In this case, the nodes may accept the malicious messages as follows:

We assume that the malicious router nodes generate the malicious MICs, and put them into the network after the key disclosure. In this case, the malicious router nodes can generate the correct MICs corresponding to the malicious messages because the message authentication key has already been disclosed. Moreover, it is not guaranteed that the valid messages corresponding to the stored MICs have reached to all of the nodes at the time of the key disclosure. So the nodes may wrongly accept the malicious messages if the nodes store the malicious MICs which are injected after the key disclosure. To prevent the wrongly acceptance, the nodes should not receive the malicious MICs. As an example, the nodes do not accept the more MICs than the specified number, which is the number of the authenticated messages per each of keys in a one-way key chain. After receiving the specified number of the MICs and replying the secure ACKs of them, the nodes dispose of the newly received MICs. Even if the malicious router nodes put the malicious MICs into the network after the key disclosure, the nodes can dispose of the malicious MICs because the nodes have already received the specified number of the MICs. The nodes do not accept malicious messages as shown in figure 5. However, this sheme lacks robustness. If an enemy injects some malicious MICs to the nodes during transmission of the specified number of the correct MICs, the nodes dispose of some correct MICs, which are received later, and do not reply the ACKs of them to the server. Therefore, the server cannot verify the receipt of the correct MICs. The authentication scheme could be stopped. In this case, the server can only detects some troubles in the network either by not having received the ACKs from the nodes or by not verifying the correctness of the received ACKs.

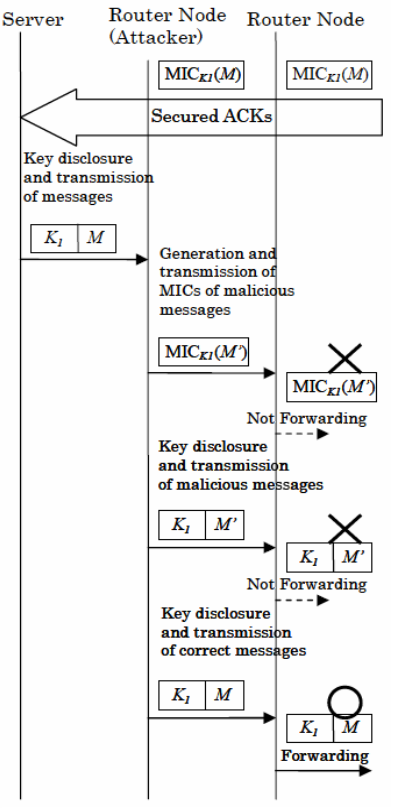

Fig. 5. Example of nodes disposing of malicious messages in the store-reduced version of the proposed scheme

Consequently, we show the result of the security consideration on both the proposed scheme and the store-reduced version of the proposed scheme in table 2.

6 Conclusion

We have proposed a reliable broadcast message authentication which is tolerant to the sender spoofing attacks even in multi-hop environments. Our schemes need no time synchronization between the sender and the receivers. Moreover, the sender acknowledges that all receivers are sure to receive the correct messages, because key disclosure of our schemes relies on the reception of secure ACKs. To minimize the number of ACKs to be received before key disclosure, we compute MICs using a Merkle hash tree-like technique. We have also proposed a way to reduce the amount of the stored information until the disclosure of the key. The proposed schemes are based on symmetric-key cryptographic primitives that are generally computationally less expensive than public key cryptographic ones, so we think our schemes are feasible for the resource-constrained sensor nodes.

However, our schemes lack robustness. Even one sensor node which has been compromised by an attacker can easily corrupt the broadcasting, and nobody can distinguish which node has been compromised and is corrupting the protocol.

Our future works are to discover the dishonest node as well as to consider other acknowledgement schemes well-suited for sensor networks.

References

1. A. Perrig, R. Canetti, J.D. Tygar, D. Song: Efficient Authentication and Signing of Multicast Streams over Lossy Channels. IEEE Symposium on Security and Privacy (2000) 56-73
2. A. Perrig, R. Canetti, J.D. Tygar, D. Song: Efficient and Secure Source Authentication for Multicast. ISOC Network and Distributed System Security Symposium (2001) 35-46
3. A. Perrig et al.: SPINS: Security Protocols for Sensor Networks. Wireless Networks Journal., vol.8, no.5 (2002) 521-534
4. R. Merkle: A Certified Digital Signature. Advances in Cryptology – Crypto'89 (1989) 218-238
5. C. K. Wong and S. S. Lam: Digital Signatures for Flows and Multicasts. IEEE/ACM Transactions on Networking, vol.7, no.4 (1999)
6. G. Gaubatz, et al.: Public key cryptography in sensor networks – revisited. 1st European Workshop on Security in Ad-Hoc and Sensor Networks, Lecture Notes in Computer Science, vol.3313, Springer, Heidelberg (2004) 2-18
7. D. J. Malan, et al.: A Public-Key Infrastructure for Key Distribution in Tiny OS Based on Elliptic Curve Cryptography. First IEEE International Conference on Sensor and Ad Hoc Communications and Networks (2004)
8. G. Gaubatz, et al.: State of the art in ultra-low power public key cryptography for wireless sensor networks. Workshop on Pervasive Computing and Communications Security – PerSec'05, IEEEE Computer Society (2005) 146-150
9. Yih-Chun Hu, et al., "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," MobiCom'02, Atlanta, Georgia, USA, (2002)