**Telecommunications and Information Exchange Between Systems**

# ISO/IEC JTC 1/SC 6

| | |
|---|---|
| **Document Number:** | N13880 |
| **Date:** | 2009-03-02 |
| **Replaces:** | |
| **Document Type:** | National Body Contribution |
| **Document Title:** | NB of China's contribution on 6N13661 Text for NP ballot, Security framework for ubiquitous sensor network |
| **Document Source:** | NB of China |
| **Project Number:** | |
| **Document Status:** | For consideration by SC 6/WG 7. |
| **Action ID:** | FYI |
| **Due Date:** | |
| **No. of Pages:** | 13 |

ISO/IEC JTC1/SC6 Secretariat Ms. Jooran Lee, KSA (on behalf of KATS)

Korea Technology Center #701-7 Yeoksam-dong, Gangnam-gu, Seoul, 135-513, Republic of Korea ;

Telephone: +82 2 6009 4808 ; Facsimile: +82 2 6009 4819 ; Email : jooran@kisi.or.kr

# Ubiquitous Sensor Network Security

**ISO/IEC JTC1/SC6 Mirror Committee, China**
**zhenhai.huang@iwncomm.com**
**2009-2**

# Table of contents

# Ubiquitous Sensor Network Security

## 1 Gernral description

Wireless Sensor Network (USN) protocol is based on the OSI model, each layer realizes a part of communication functions and offers services to the high-level layer.

### 1.1 Physical layer

The physical layer defines the physical radio channel and the interfaces between the Medium Access Control (MAC) sublayer and the physical layer to provide the physical layer data services and the physical layer mamagement services. The physical layer data services send and receive data in the wireless physical channel and the physical layer management services maintain a database consisting of physical layer data.

### 1.2 MAC sublayer

The MAC sublayer provides two services: MAC layer data services and MAC layer management services, the former guarantees correct transmission of the MAC protocol data unit in the physical layer data services, the later maintains a database which stores the MAC sublayer protocol-related information.

The main functions of the MAC sublayer include: beacon management, channel access，Guaranteed Time Slot ( GTS ) management, frame confirmation, sending confirmation frame, connection and disconnection.

### 1.3 Network layer

### 1.4 Application layer

## 2 Offline key predistribution and establishment of the preshared key

The keys used in the security mechanisms are based on the pre-distribution keys before deployment. According to the usage of the keys, the following key pre-distribution methods are defined.

### 2.1 Random-pairwise key predistribution and establishment

#### 2.1.1 Key predistribution scheme

The random-pairwise keys scheme is a modification of the pairwise keys scheme based on the observation that not all ($n$-1) keys need to be stored in the node's key ring to have a connected random graph with high probability. Erdős–Rényi 's formula allows us to calculate the smallest probability $p$ of any two nodes being connected such that the entire graph is connected with high probability $c$. To achieve this probability $p$ in a network with $n$ nodes, each node need only store a random set of $n*p$ pairwise keys instead of exhaustively storing all ($n$-1). Reversing the calculation, if a node can store $m$ keys, then the maximum supportable network size is $n = m/p$. Depending on the model of connectivity, $p$ may grow slowly with $n$ when $n$ is large (intuitively,

*p* cannot decrease as *n* goes toward infinity, since it is more likely that a large graph is disconnected than a smaller graph). Hence, *n* should increase with increasing *m* and decreasing *p*. The exact rates will depend on the deployment model.

In the pre-deployment initialization phase, a total of $n = m/p$ unique node identities are generated. The actual size of the network may be smaller than *n*. Unused node identities will be used if additional nodes are added to the network in the future. Each node identity is matched up with *m* other randomly selected distinct node IDs and a pairwise key is generated for each pair of nodes. The key is stored in both nodes' key rings, along with the ID of the other node that also knows the key.

### 2.1.2 Key establishment

#### 2.1.2.1 Link key establishment

During the establishment procedure of the network, the node first broadcast its ID to it's neighbor nodes in the scope of one-hop communications, the neighbor node determines whether it has a shared key with the broadcast node through searching in the ID lists, if there exists a shared key, the security connection can be established, the shared key is known as link key. Otherwise, the two nodes are concidered as no connection. If the neighbor nodes with the shared key require to authenticate each other, the authentication procedure will be started. The link key is used for the protection of the MAC layer frame.

#### 2.1.2.2 Network key establishment(Path key)

Nodes can set up path keys with nodes in their vicinity whom they did not happen to share keys with in their key rings. If the graph is connected, a path can be found from a source node to its neighbor. The source node can then generate a path key use of the two nodes ID and send it securely via the path to the target node. We denote path keys as network key, it mainly use for protection of network layer frames and application layer frames.

#### 2.1.2.3 Group key establishment

The group key establishment method is designed according to the network topology.

## 2.2 Broadcast authentication key predistribution and establishment

### 2.2.1 Key predistribution scheme

This section gives the distribution methods of Broadcast authentication key. The methods make an improvement of μTESLA by introducing the security mechanism of One Way Chain and Merkle Tree. It constructs μTESLA Parameters Hash Chain (μTPC) to distribute and authenticate the initial parameter of μTESLA. Then it constructs μTPC Merkle Tree (μTPCT) to distribute and authenticate the initial parameter of μTPC.

#### 2.2.1.1 Construction of μTPC

The essential problem to scaling up μTESLA is how to distribute and authenticate the initial μTESLA parameters (we'll call it μTP later), mainly including the key chain commitments, starting time, duration of each time interval, etc. The multi-level μTESLA uses high-level μTESLA instances to authenticate the parameters of low-level ones. It inherits the authentication delay introduced by μTESLA during the distribution of those parameters. The consequence of such authentication delay is that an attacker can launch DoS attacks to disrupt the distribution of initial μTESLA parameters. Moreover, multi-level μTESLA cannot handle a large number of senders. Tree-based μTESLA protocol uses Merkle Tree mechanism to distribute μTP. Using the certificate from Merkle tree, receiver nodes can authenticate μTP immediately, so it can resist DoS attacks. But the cost of Tree-based μTESLA is too large. The μTPCT-based broadcast authentication protocol proposed in this paper constructs μTPC to distribute and authenticate μTP. It can resist DoS attacks and only need small cost.

In sensor networks with multiple BNodes, in view of the task to be performed, BNode may have different characteristics. We entitle the life cycle, broadcasting frequency and real-time requirement of BNode as its characteristic parameter, for short as FP. BServer will construct μTP based on the FP of BNode. For example, for the BNode with short life cycle, high broadcasting frequency and strong real-time requirement, BSever will construct a special μTP which contains short key disclosure lag and less μTESLA instances with short time interval. FP can be expanded as required.

μTPC is composed of μTP and One Way Chain. After FP is determined, BServer will firstly divides the lifetime of BNode into $N$ time intervals with length of $T_N$, such that the duration of $T_N$ (e.g., 30 minutes) is suitable for running a μTESLA instance on a BNode and sensor nodes efficiently. According to broadcasting frequency and real-time requirement of BNode, BServer will divide $T_N$ into $n$ time interval with length of $T_n$. Bases on $N$ and $n$, BServer use pseudo-random function F to generate $N$ μTESLA key chains which linked together. At first, BServer generates the last key $K_{N,n}$ of the $N$-th μTESLA key chain at randomly. Then using hash function H (e.g., SHA-1), BServer generate rest keys of the $N$-th μTESLA key chain according to $K_{N,i}=H(K_{N,i+1})$. For the ($i$-1)-th μTESLA key chain, BServer generates the last key by performing a pseudo random function on the first key (the key next to the commitment) of the $i$-th μTESLA key chain. Then generates rest keys of ($i$-1)-th μTESLA key chain by performing H on it's last key. By this way, BServer generate all μTESLA key chains till to the last one. Figure 1 shows the construction of μTESLA key chains.
After all μTESLA key chains was generated, for the BNode $j$, BServer will assigning different keys to different time intervals $T_n$. Accordingly, there will come into being $N$ μTESLA instances. Where the initial parameter of the $i$th μTESLA instance is $\mu TP_i=\{T_s \parallel K_{i,0} \parallel T_i \parallel T_{int} \parallel d\}$, where $T_s$ denotes current time, $K_{i,0}$ denotes the commitment, $T_i$ denotes starting time, $T_{int}$ denotes synchronization interval, $d$ denotes disclosure lag of the key. After all μTP were determined, BServer generate a value $U_N$ randomly, then compute each $U$ value by $U_{i-1}=H(U_i \parallel \mu TP_{i-1})$ till to $U_0$, " ∥ " denotes message concatenation. Finally, BServer constructs a μTPC which including $N$ μTPs. Figure 2 shows an example of construction of μTPC.

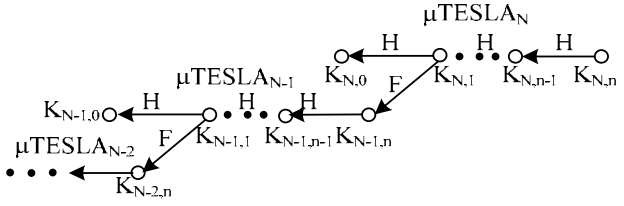


Figure 1. Construction of μTESLA key chains of μTPC, in which each $K_{i,n}$ is derived from $K_{i+1,1}$ using a pseudo random function F.

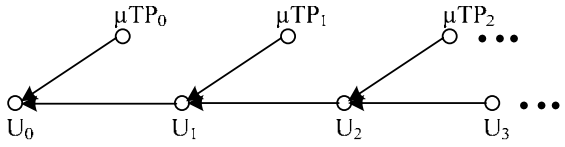


Figure 2. Construction of μTPC, where Ui = H (Ui+1||μTPi).

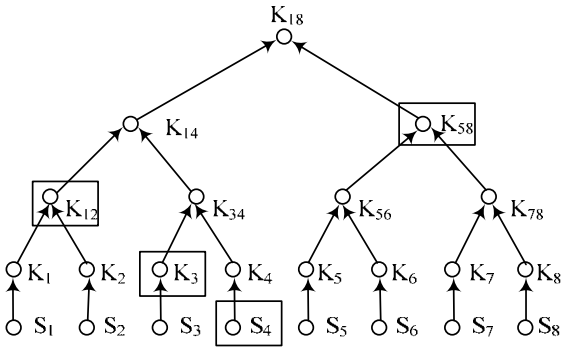


Figure 3. A μTPCT tree with 8 leaf nodes, the nodes in the boxes constitutes the certificate of S4, we call it PCert4.

#### 2.2.1.2 Construction of μTPCT

Suppose there exists $m$ BNodes in sensor networks. For convenience, we assume $m = 2^k$, where $k$ is an integer. Before deployment, BServer pre-computes $m$ μTPC, each of which is assigned a unique, integer-valued ID between 1 and $m$. For the sake of presentation, we denote the $j$-th U value of $i$-th μTPC as $U_{i,j}$, the $j$-th μTP as $\mu TP_{i,j}$, the $i$-th initial parameter (including $U_{i,0}$, $ID_i$) of μTPC as $S_i$. BServer then computes $K_i$=H($S_i$) for all $i \in \{1,...,m\}$.Then, it constructs a Merkle tree using $\{K_1,...,K_m\}$ as leaf nodes. Each non-leaf node is computed by applying H to the concatenation of its children nodes. We call such a Merkle tree as μTPCT. Figure 3 shows a μTPCT with eight μTPC, where $K_{12}$=H($K_1 \parallel K_2$), $K_{14}$=H($K_{12} \parallel K_{34}$), $K_{18}$=H($K_{14} \parallel K_{58}$), etc.

BServer also constructs a parameter certificate for each μTPC instance. The certificate for $i$-th μTPC instance consists of $S_i$ and the values corresponding to the siblings of the nodes on the path from $i$-th leaf node to the root of μTPCT. For example, the parameter certificate for the 4th μTPC instance in Figure 3 is $PCert_4=\{S_4,K_3,K_{12},K_{58}\}$. For each BNode which will use a given μTPC instance, BServer distributes the μTPC and the corresponding parameter certificate to it. BServer also pre-distributes the root of the μTPCT to all potentially receivers of broadcast messages.

Before construction of μTPCT, if there are same parts in all μTP of some μTPC, we will take the same parts of μTP together with initial parameter of μTPC as leaf nodes to construct μTPCT. For example, if the disclosure lag $d$ and synchronization intervals $T_{int}$ in all μTP of $\mu TPC_i$ are same, we will take $T_{int}$, $d$ together with initial parameter of $\mu TPC_i$ as leaf nodes to construct μTPCT. Accordingly, the same parts of μTP will distributed together with the certificate of μTPC with only one time. In the process of μTP distribution, BNode need to distribute the discrepant part only. By this way, substantive communication cost will be saved.

### 2. 2. 2 Key establishment

There is no need for special key establishment procedure, only updating the key in time is needed.

### 2. 3 Other keys predistribution Scheme and key establishment

Other key predistribution schemes can be added.

## 3 Cryptographic mechanisms and authentication mechanisms

The security mechanisms is composed of the authentication mechanisms and cryptographic mechanisms. The authentication mechanisms are mainly used to validate the legitimacy of the equipement when it entering to the network, to ensure that the equipements in the network are legitimate and credible. The cryptographic mechanisms are mainly used for processing the transmitted frame to ensure the data security.
The specific authentication method of the authentication mechanisms is realized by the authentication suite.

### 3. 1 Cryptographic mechanisms

The cryptographic mechanisms are based on the symmetric cryptosystem, the method for establishing and maintaining of the keys are in accordance with the key pre-distribution suite, the recommended methods of establishment and maintenance of keys can be found in Annex F.

The cryptographic mechanisms provide the following given combinations of security services:

a) Data confidentiality: to ensure that the message transmitted only can be seen by the destination object;
b) Data authenticity: to confirm the source of the message(and therefore to confirm the message has not been alterd in the transmission);
c) Reply attack protection: to ensure that the duplication of the message was denied.

The provided frame protection can be different, and different levels of data authenticity and optional data confidentiality are allowed ,.If necessary, the retransmission protection can be provided.

The protection of the frame key can make use of the shared-key between the two peer devices(link key or network key),also the shared-key among a set of devices(group key)can be used, if the group key is used for the communication between two peer devices, protection is provided only against outside devices and not against potential malicious devices in the key-sharing group.

### 3. 2 Authentication mechanisms

#### 3. 2. 1 Preshared Key authentication

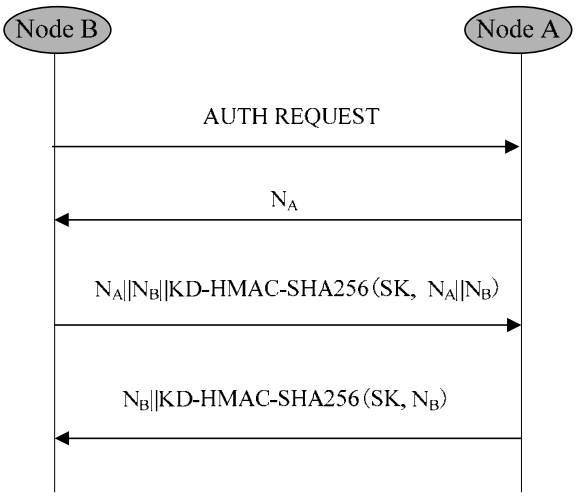


Figure 4. Preshared Key authentication procedure

In the following text, the PSK denotes the pre-shared key between A and B, the "$ADDID_x$" denotes the concatenation of A's address and B's address. The mechanism is performed as follows:

**Step 1**: B sends an "AUTH REQUEST" to A to start the authentication procedure;

**Step 2**: When A received the message, A generates a nonce $N_A$ and sends it to B;

**Step 3**: B generates a nonce $N_B$, calculates the session key SK=KD-HMAC-SHA256(PSK,$ADDID_1||N_B||N_A||$"pairwise key expansion for unicast and additional keys and nonce") and a message authentication code $MAC_1$=KD-HMAC-SHA256(SK, $N_B||N_A$), then B sends $MAC_1$,$N_B$ and $N_A$ to A.

**Step4**: A checks $N_A$ firstly, then calculates SK=KD-HMAC-SHA256(PSK, $ADDID_2||N_B||N_A||$"pairwise key expansion for unicast and additional keys and nonce"), and $MAC_2$=KD-HMAC-SHA256(SK, $N_B||N_A$). If $MAC_1$= $MAC_2$, A sends $MAC_3$=KD-HMAC-SHA256(SK,$N_B$) and $N_B$ to B.

### 3. 2. 2 ID-based authentication

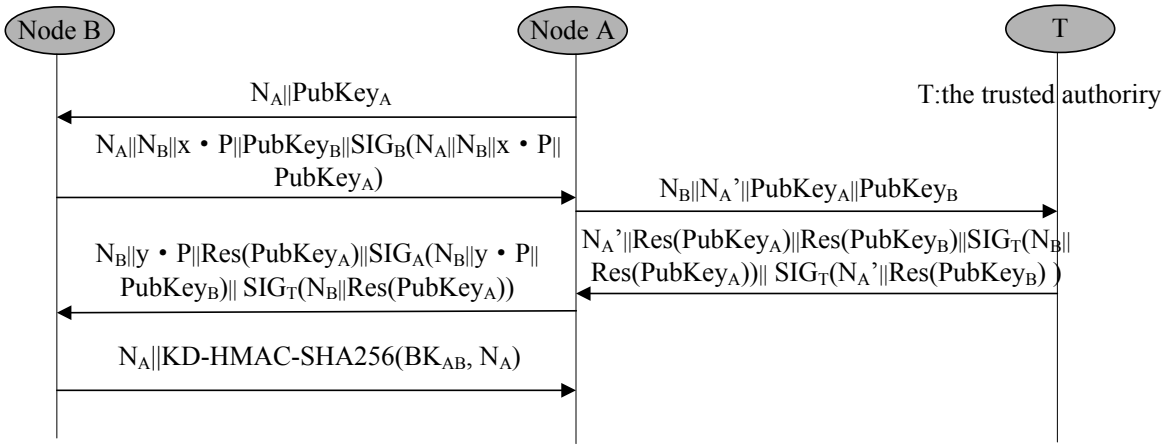


Figure 5. ID-based authentication procedure

In the following text, $PubKey_X$ denotes the public key of X. $SIG_X$ denotes the signature of X. Res ($PubKey_X$)

denotes the result of verification of $PubKey_X$. The mechanism is performed as follows:

**Step 1**: A generates a random number $N_A$ and sends $N_A||PubKey_A$ to B;

**Step 2**: B generates a temporal secret key x and a temporal public key x·P for ECDH, then sends$N_A||N_B||x \cdot P||PubKey_B||SIG_B(N_A||N_B||x \cdot P||PubKey_A)$ to A.

**Step 3**: After A received the message from B, A generates another ramdom number $N_A$' and sends $N_B||N_A'||PubKey_A||PubKey_B$ to T;

**Step 4**: On receipt of the message from A, T inspects the validity of $PubKey_A$ and $PubKey_B$, and sends $N_A'||Res(PubKey_A)||Res(PubKey_B)||SIG_T(N_B||Res(PubKey_A))||\ SIG_T(N_A'||Res(PubKey_B))$ to A.

**Step 5**: After A received the message from T, A verifies $N_A$' firstly, then verifies $SIG_T(N_A'||Res(PubKey_B))$ by checking if $N_A$' agrees with the one sent to T. Then A generates a temporal secret key y and a temporal public key y·P for ECDH, calculates $BK_{AB}$=KD-HMAC-SHA256((x·y·P)abscissa, NA||NB|| "base key expansion for key and additional nonce"), and sends $N_B||y \cdot P||Res(PubKey_A)||SIG_A(N_B||y \cdot P||PubKey_B)||\ SIG_T(N_B||Res(PubKey_A))$ to B.

**Step 6**: B verifies $N_B$ firstly, then verifies $SIG_T(N_B||Res(PubKey_A))$ and $SIG_A(N_B||y \cdot P||PubKey_B)$ by checking $N_A$ and $N_B$ respectively. Then calculates the Base Key $BK_{AB}$=KD-HMAC-SHA256((x·y·P)abscissa, NA||NB|| "base key expansion for key and additional nonce"). Basing on $BK_{AB}$, B calculates MAC= KD-HMAC-SHA256($BK_{AB}$, $N_A$) and then sends it with $N_A$ to A. A verifies the MAC from B basing on $BK_{AB}$.

## 4 USN Security specificaiton

This section involves only the network layer and application layer security specifications. The MAC layer security refers to the MAC layer protocol.

### 4.1 Overview

The security services provided by the USN include: key establishment,authentication, frame protection as well as node management. These services constitute an integral part of the security policy of USN nodes. This chapter is a detailed description of the usage and the basic functions of these security services.

#### 4.1.1 Network layer security

When the network layer frame in need of protection, The frame protection mechanisms must be enabled in the node. The specific frame protection strategy is determined by the appointed security level.

#### 4.1.2 Application layer security

When the application layer frame in need of protection, the frame protection mechanisms must be enabled in the node. The specific frame protection strategy is determined by the appointed security level. The application layer also provides key management, node management, authentication and other security services.

## 4.2 Network layer security

The network layer processes the outgoing and incoming frames which need to be protected. The safety operation of the network layer is controlled by the upper layer through establishing a suitable key and frame couter as well as setting up a security level.

### 4.2.1 Frame security

#### 4.2.1.1 Key

The keys used for protecting the network layer frame are provided by the upper layer.

#### 4.2.1.2 Security processing of outgoing frames

##### 4.2.1.2.1 Encapsulation of non-broadcast frames

The non-broadcast frames include unicast and multicast frames, which using the same frame protection mode as the MAC layer do.

##### 4.2.1.2.2 Encapsulation of the broadcast frames

The broadcast frame use the message authentication code (MAC) for protection to ensure that the broadcast message is from the legitimate broadcaster and has not been tampered during the transmission. The MAC is calculated by the broadcast authentication key, and the key will be sent to the receiver after a pre-defined delay.

#### 4.2.1.3 Security processing of incoming frames

##### 4.2.1.3.1 Decapsulation of non-broadcast frames

The decapsulation of the non-broadcast frame is resolved according to the security level of the incoming frames.

##### 4.2.1.3.2 Decapsulation of the broadcast frame

The protected broadcast frames will be stored when received. The broadcast authentication key will be verified by the existing key which came from the broadcast authentication key chain after received. Then the broadcast frame will be authenticated basing on the broadcast authentication key.

### 4.2.2 The format of the safety frame

According to the network layer frame format, the security control domain field is added to constitute the safety network layer frame.

### 4.2.3 Security-related NIB (Network layer Information Base)attributes

It refers to the security-related NIB attribute.

### 4.3 Application layer security

The network layer is responsible for the processing of the outgoing and incoming frames which need to be protected, as well as key management and node management.

#### 4.3.1 Key management services

In this section, the key management refers to the management of the pre-distributed key.

##### 4.3.1.1 Key establishment

The key establishment procedure can be found in Clause 2.

##### 4.3.1.2 Key maintenance

#### 4.3.2 Entity authentication services

The specific authentication mechanisms is realized by the authentication suite, see subclause 3.2. In the ID-based authentication procedure, the nodes use ECDH exchange to negociate a base key BK; in the preshared key authentication procedure, the nodes using their shared key as a seed to export the base key BK. Then, in the unicast key negociation procedure, the nodes exchange respectively a random number between them and export the unicast session key based on BK. Finally, in the multicast key announcement procedure, the nodes expand respectively the NMK(notification master key) to generate the multicast session key(the multicast key is only used by the sink node).

#### 4.3.3 Frame security

##### 4.3.3.1 Key

The safety of outgoing and incoming frame is ensured by the key produced by the authentication services.

##### 4.3.3.2 Security processing of outgoing frames

It uses the same frame protection mode as the MAC layer do.

##### 4.3.3.3 Security processing of incoming frames

It resolves according to the security level of the received frame.

#### 4.3.4 Command frames

It defines the format of the command frames for various security services.

#### 4.3.5 Node management services

##### 4.3.5.1 Node update

It provides a safe way to inform a node the status updating information of other nodes.

##### 4.3.5.2 Node delete

It provides a safe way to inform a node to delete information of other nodes.

#### 4.3.6 AIB (Application Information Base) security-related attribute

It defines the security-related attributes which related to the AIB.

### 4.4 Public safety element

The format of the head of the security frame,including security control field, frame counter,source address, key serial number,etc.

### 4.5 Functional description

This sub-clause provides detailed descriptions of how the security services shall be used in a USN.