Paper

How the Role-Based Trust Management Can Be Applied to Wireless Sensor Networks

Anna Felkner

Research and Academic Computer Network (NASK), Warsaw, Poland

Abstract—Trust plays an important role in human life environments. That is why the researchers has been focusing on it for a long time. It allows us to delegate tasks and decisions to an appropriate person. In social sciences trust between humans was studied, but it also was analyzed in economic transactions. A lot of computer scientists from different areas, like security, semantic web, electronic commerce, social networks tried to transfer this concept to their domains. Trust is an essential factor in any kind of network, whether social or computer. Wireless sensor networks (WSN) are characterized by severely constrained resources, they have limited power supplies, low transmission bandwidth, small memory sizes and limited energy, therefore security techniques used in traditional wired networks cannot be adopted directly. Some effort has been expended in this fields, but the concept of trust is defined in slightly different ways by different researchers. In this paper we will show how the family of Role-based Trust management languages (RT) can be used in WSN. RT is used for representing security policies and credentials in decentralized, distributed access control systems. A credential provides information about the privileges of users and the security policies issued by one or more trusted authorities.

Keywords—access control, role-based trust management, trust, wireless sensor networks.

1. Introduction

The concept of trust can be understood in quite various ways. Generally it can be based on personal experience, reputation or recommendation. A lot of work connected with trust has been done by sociologist, economists, psychologists and lately also by computer scientist. It has become very important in the late years as a consequence of the growth of fields such as Internet transactions or electronic commerce.

Establishing trust in a network gives two important benefits:

- it helps to make traditional security mechanisms more robust and reliable,
- it can solve the problems that can not be solved through traditional security mechanisms.

Wireless sensor networks are becoming increasingly important due to the growing range of their capabilities. The range of applications of WSN is so wide that it tends to invade our everyday life. The services offered by wireless sensor networks can be classified into four major categories: monitoring, alerting, providing information, and actuating. Their significance is more and more important, especially in relation to gathering information, in fields such as health care, defence, environmental and structural monitoring, homeland security, industry control, intelligent green aircrafts, smart roads and others. There are many applications which are intended to monitor physical and environmental phenomena, such as ocean and wildlife, pollution, earthquakes, and water quality. The main purpose of these WSNs is to provide physical information such as temperature, light, radiation, and others to a computer system and it offers efficient solutions in a great variety of application domains. The network can modify the state of an external system (e.g., barriers, traffic lights, irrigation system) according to the data, going beyond its sensing capabilities. In the future, a sensor network will survey our health, home, the roads we follow, the office or the industry we work in or even the aircraft we use, in an attempt to enhance our safety.

It is a quite young technology with many interesting research problems. One of the issues is security, and trust is a part of it. Very often applications which use WSN require high dependability. Also, networks which provide more sophisticated services require more effective security mechanism. Unfortunately, not all security solutions suitable for traditional networks are appropriate for WSN, because of their resource constraints.

Traditional trust management schemes that have been developed for wired and wireless ad-hoc networks are not well suited for wireless sensor networks, due to their higher consumption of resources such as memory and power. The sensor nodes are highly constrained in terms of communication bandwidth, processing resources, computational capabilities, memory space, and battery capacity.

Some of the approaches adopted for WSN try to imitate those for ad-hoc or peer-to-peer networks, but this is not always possible due to the difference in the features of these networks (like the computational power, energy-constraint and also the size of the networks). In this work we will try to show how our approach to the concept of trust management can be adjusted to wireless sensor networks.

There are some works connected with trust used in sensor networks to increase their security and reliability. Most of these works are based, or take into consideration, the concept of reputation. Quite often the trust is obtained as a function of reputation. Reputation is the opinion of one person about the other, in WSN it can be the opinion of

one node about another. It can be built over time based on the history of behavior of the node.

Guaranteeing that confidential data and services offered by a computer system are not made available to unauthorized users is an increasingly significant and challenging issue, which must be solved by reliable software technologies that are used for building high-integrity applications. The data, whether in electronic, paper or other form must be properly protected. The traditional solution to this problem is access control techniques by which users are identified, and granted or denied access to a system, data and other resources, depending on their individual or group identity. This approach fits well into closed, centralized environments in which the identity of users is known in advance. However, access control in such a resource constrained WSN provides significant challenges, and in an ongoing area of research and trust management is a specific kind of access control in which decisions are based on credentials issued by multiple principals.

The paper is organized as follows: An overview of the work related to trust management in wireless sensor networks is given in Section 2. Section 3 shows the overview of the family of Role-based Trust management languages, including syntax and inference system over *RT* credentials. Section 4 describes time validity in *RT* languages with inference system. Final remarks are given in Conclusions.

2. Related Work

Trust has been the focus of researchers for a long time. It origins derives from social sciences where trust between humans was studied. Since Marsh [1] introduced a computational model for trust in his thesis, trust mechanism has gradually obtained more and more researcher's ([2], [3], and so on) interest for its flexibility and extensibility. Numerous trust models were proposed in social network, distributed network, peer-to-peer computing, ad-hoc network, and so on.

Although intuitively easy to conceive, the notion of trust has not been formally defined unanimously. Trust in wireless sensor network is yet to adopt a formal definition. A dictionary definition states that trust is a belief or confidence in the honesty, goodness, skill or safety of a person, organization or thing [4]. It means that such a belief is based on explicit assessment of trustworthiness of the trusted party.

There is a large diversity in the understanding of the concept of trust. The concept of trust management in distributed systems was first defined in 1996, and the approach presented by authors of this paper is based on this definition. Along with the notion of trust, comes that of reputation, which is the opinion of one person about the other, of Internet buyer about an Internet seller, and one node in wireless sensor network about another. Also, reliability is connected with the trust concept. It was originally a measure of how long a machine can be trustworthy. Trust can be understood as a derivation of the reputation of an entity. Based on a reputation, a level of trust is granted upon an entity. The reputation itself has been built over time based on that entity's history of behavior, and may be reflecting a positive or negative assessment.

There are not many publications connected with the area of trust management systems for wireless sensor networks. Most of the work in this field has been made in the last few years (e.g., Reputation-based Framework for Sensor Networks (RFSN) [5], Agent-based Trust and Reputation Management (ATRM) [6], and Parameterized and Localized Trust Management Scheme (PLUS) [7]). However, big efforts have been made in related areas such as introducing the concept of trust management schemes for increasing security and reliability in peer-to-peer networks [8], [9] and ad-hoc networks [10]–[16]. There are some other works available in the literature, e.g., [17]–[22], and so forth, that discuss trust in WSN but not in much detail.

Very often in the literature, trust has been used in WSNs for assessing the availability, reliability, or security property of a node (e.g., whether a node is malicious or not) based on past interaction experiences [5], [23].

Ganeriwal *et al.* [5] were among the first who defined comprehensive trust management scheme for sensor networks. They propose a reputation-based framework for high integrity sensor networks based on a bayesian formulation (more specifically, a beta reputation system) where nodes maintain reputation for other nodes, and use it to evaluate their trustworthiness. The architecture of the framework consists of a watchdog mechanism, reputation, second hand information, trust, and behavior. In this framework each sensor node maintains reputation metrics which both represent past behavior of other nodes, and are used as an inherent aspect in predicting their future behavior. Reputation is stored in a table where the entries are built by the nodes through the watchdog mechanism. Nodes not only use their own direct observations, but they also exchange information with other nodes (second hand information). Reputation is calculated by using the beta reputation distribution and trust is obtained as a function of reputation. Then the behavior of a node is given according to whether the trust values are respectively above or below a given threshold.

A watchdog mechanism is also used in Chen *et al.* [17]. In their work reputation is similarly used in order to define a trust management system for wireless sensor networks. Their model uses probability, statistics and mathematical analysis. They consider the concept of certainty for trust. The first-hand information is aggregated by using a watchdog mechanism. A reputation space is defined considering the positive and negative outcomes, and trust space is defined from the reputation space. In [13] reputation is also considered as a way for building trust.

In [22] Shaikh *et al.* propose a lightweight group based trust management scheme (GTMS) for distributed wireless sensor networks in which the whole group will get a single trust value. Instead of calculating individual trust, in some cases it is much more appropriate to calculate the trust for the entire group. GTMS uses a hybrid trust management scheme instead of using centralised or distributed schemes,

which helps in keeping minimum resource utilisation at the sensor nodes.

In [24] Yao *et al*. introduce a framework similar to existing approaches for ad-hoc networks where trust values are assigned to each node. A trust evaluation process is performed based on the localised trust model and two kinds of knowledge: personal reference gained by interaction with the evaluated node (suspect node) and reputation sent by the juries (specific nodes).

Yao *et al*. [7] also propose a parametrised and localised trust management scheme for WSN security, especially for secure routing where each node maintains highly abstracted parameters to evaluate its neighbours.

Aivaloglou and Gritzalis [23] show a hybrid trust and reputation management protocol for WSNs by combining certificate-based and behaviour-based trust evaluations.

Zhiying *et al*. [25] find distributed trust models appropriate for large-scale sensor network security design, because each node focuses on the trustworthiness of its neighbours and can assess if these nodes comply with agreed security policies. Authors propose an appropriate security framework with different security schemes. Unfortunately, their work does not take into consideration the resource limits of nodes in sensor networks.

Zia in [26] proposes a security framework where integrating the reputation and trust management mechanism is used to provide a comprehensive security solution against well-known threats. In this work nodes monitor their neighboring nodes and rank the neighbors to execute a trust vote.

Momani *et al*. [27] also introduce a trust model and a reputation system for WSNs based on sensing continuous data.

Chen *et al.* [28] propose a distributed agent-based trust management scheme where each agent node monitors the behavior of the nodes within its radio range, and broadcasts their trust ratings.

As it was shown just above, there is a large diversity in the understanding of the concept of trust, also in wireless sensor networks. The term trust management was first applied in the context of distributed access control in [2] and the approach presented here is based on this definition.

Traditional access control systems usually rely on Role-Based Access Control model [29], [30] which groups the access rights by the role name and limits the access to a resource to those users who are assigned to a particular role.

The first trust management application described in the literature was PolicyMaker [31] which defined a special assertion language capable of expressing policy statements, which were locally trusted, and credentials that had to be signed using a private key. The next generation of trust management languages were KeyNote [32], which was an enhanced version of PolicyMaker, SPKI/SDSI [33] and a few other languages [34]. All these languages allowed assigning privileges to entities and used credentials to delegate permissions from its issuer to its subject. What was missing in those languages was the possibility of delegation based on attributes of the entities and not on their identity.

Responding to this need, a family of Role-based Trust management languages has been introduced in [35]–[38], and practical application using the *RT* language to control access to virtual machines was presented in [39]. These languages have a well-defined syntax and semantics, which made them easy to extend in order to apply them to different needs. One of the extensions is the use of time validity constraints of the credentials, which made the languages of the RT family more realistic, because in the real world permissions are usually given just for a limited period of time. Time-dependant credentials were introduced in [40] but only for RT_0 language. Because RT^T language is more complex, powerful and allows to express security policies more suited to real needs, we decided to develop extensions to this specific language, which has not been done before. The complex time-dependant inference system with necessary proofs was introduced in [41].

3. Role-Based Trust Management

Role-Based Access Control (RBAC) model [29], [30] is the most flexible type of access control policy. It uses user role to control which users have access to particular resources. Access rights are grouped by the role name and access to resources is restricted to the users who are assigned to appropriate roles. The meaning of roles in RT captures the notion of groups of users in many systems and has been borrowed from RBAC approach. This type of access control works well in a large-scale centralized system and is often used in enterprise environments. Quite different challenges arise in decentralized and open systems where the identity of users is not known in advance and the set of users can change. It is also different in a wireless sensor network where sets of sensors can change rapidly. The identity of a user itself does not help in making decisions about their rights. What is needed to make such decisions is information about the privileges assigned to the user by other authorities, as well as trust information about the authority itself.

The term of *trust management* was introduced in 1996 by Blaze et al. in [2] who defined it as a unified approach to specify and interpret security policies, credentials and trust relationships. In a trust management system an entity's privilege is based on its attributes instead of its identity. An entity's attributes are demonstrated through digitally signed credentials issued by multiple principals. A *credential* is an attestation of qualification, competence or authority issued to an individual by a third party. Examples of credentials in real life include identification documents, driver's licenses, membership cards, keys, etc. A credential in a computer system can be a digitally signed document. Such a concept of trust management has evolved since that time to a much broader context of assessing the reliability and developing trustworthiness for other systems and individuals [42]. In this paper, however, we will use the term trust management only in a meaning restricted to the field of access control.

The potential and flexibility of trust management approach stems from the possibility of *delegation*: a principal may transfer limited authority over a resource to other principals. Such a delegation is implemented by means of an appropriate credential. This way, a set of credentials defines the access control strategy and allows deciding on who is authorized to access a resource, and who is not. The concept of delegation can also be used in a WSN, especially in routing structures that is why we will try to show how the permissions can be delegated from one sensor to another.

RT languages combine trust management and RBAC features. To define a trust management system, a language is needed for describing entities (principals and requesters), credentials and roles which the entities play in the system.

The core language of RT family is RT_0, described in detail in [37]. It allows describing localized authorities for roles, role hierarchies, delegation of authority over roles and role intersections. All the subsequent languages add new features to RT_0, they are progressively increasing in expressive power and complexity. RT_1 introduces parameterized roles, which can represent relationships between entities. RT_2 extends RT_1 with logical objects, which can be used to represent permissions given to entities with respect to a group of logically related objects (resources). These extensions can help in keeping the notation concise, but do not increase the expressive power of the language, because each combination of parameters in RT_1 and each permission to a logical object in RT_2 can be defined alternatively as a set of separate roles in RT_0. The most powerful language in the family is RT^T, as it provides useful capabilities not found in any other languages: manifold roles to achieve both agreement of multiple principals from one set and from disjoint sets and role-product operators, which can express threshold and separation of duties policies. Similar to a role which defines a set of principals a manifold role defines a set of principal sets, each of which is a set of principals whose cooperation satisfies the manifold role. A threshold policy requires a specified minimum number of entities to agree on some fact, i.e., it requires agreement among k out of a set of entities that satisfy a specified condition, e.g., in a requirement that two different bank cashiers must authorise a transaction. Separation of duties policy requires a set of entities, each of which fulfils a specific role, to agree before access is granted.

RT^D provides mechanism to describe delegation of rights and role activations, which can express selective use of capacities and delegation of these capacities. In many scenarios, an entity prefers not to use or delegate all his rights. For example, if an entity D activates the role $A.r$ to use it in a session B, it can take the form of delegation credential, as a:

$$D \xrightarrow{D \text{ as } A.r} B,$$

where D *as* $A.r$ is called a role activation. B can further delegate this role activation to C by issuing the credential,

$$B \xrightarrow{D \text{ as } A.r} C.$$

An entity can issue multiple delegation credentials to another entity and also, several role activations can be delegated in one delegation credential.

The features of RT^T and RT^D can be combined together with the features of RT_0, RT_1 or RT_2. A more detailed treatment of RT family can be found in [36].

The languages have a precise syntax and semantics definition. A set-theoretic semantics, which defines the meaning of a set of credentials as a function from the set of roles into the power set of entities, has been defined for RT_0 [40], [37] and we defined relational semantics which apply also to other members of the family up to RT^T in [43]. The logic-programming semantics of RT_0 credentials was first introduced in [36], a modified version of this semantics was shown in [40] and the semantics of all the other languages up to RT^T was described in [44]. The member sets of roles can also be calculated in a more convenient way using an inference system which defines an operational semantics of RT languages. An inference system consists of an initial set of formulae that are considered to be true, and a set of inference rules that can be used to derive new formulae from the known ones. The operational semantic was described in [45] and [40].

Table 1
Supported features of RT languages

RT language	Supported features
RT_0	– localized authorities for roles, – role hierarchies, – delegation of authority over roles, – attribute based delegation of authority, – role intersections.
RT_1	features of RT_0 plus: – parameterized roles, – attribute-relationship based delegation, – attribute-field constraints.
RT_2	features of RT_1 plus: – logical objects.
RT^T	features of RT_0 plus: – manifold roles, – threshold policies, – separation-of-duty policies.
RT^D	features of RT_0 plus: – selective use of role membership, – dynamic credential delegation.

A summary of the features supported by particular RT languages is shown in Table 1.

3.1. The Syntax of RT Family Languages

Basic elements of RT languages are entities, role names, roles and credentials. *Entities* represent principals that can define roles and issue credentials, and requesters that can make requests to access resources. An entity can, e.g., be a person or program identified by a user account in a com-

puter system or a public key. *Role names* represent permissions that can be issued by entities to other entities, or groups of entities. *Roles* represent sets of entities that have particular permissions granted according to the access control policy. *Credentials* define roles by appointing a new member of the role or by delegating authority to the members of other roles.

There are six types of credentials in RT^T (first four can also be used in RT_0, RT_1, and RT_2) which are interpreted in the following way:

- $A.r \leftarrow B$ – *simple membership*: entity B is a member of role $A.r$.
- $A.r \leftarrow B.s$ – *simple inclusion*: role $A.r$ includes (all members of) role $B.s$. This is a delegation of authority over r from A to B, because B may cause new entities to become members of the role $A.r$ by issuing credentials that define $B.s$. The hierarchy of roles is also possible.
- $A.r \leftarrow B.s.t$ – *linking inclusion*: role $A.r$ includes role $C.t$ for each C, which is a member of role $B.s$. This is a delegation of authority from A to all the members of the role $B.s$. The expression $B.s.t$ is called a *linked role*.
- $A.r \leftarrow B.s \cap C.t$ – *intersection inclusion*: role $A.r$ includes all the entities who are members of both roles $B.s$ and $C.t$. This is a partial delegation from A to B and C. The expression $B.s \cap C.t$ is called an *intersection role*.
- $A.r \leftarrow B.s \odot C.t$ – role $A.r$ can be satisfied by a union set of one member of role $B.s$ and one member of role $C.t$. A set consisting of a single entity satisfying the intersection role $B.s \cap C.t$ is also valid.
- $A.r \leftarrow B.s \otimes C.t$ – role $A.r$ includes one member of role $B.s$ and one member of role $C.t$, but those members of roles have to be different entities.

3.2. Inference System over RT Credentials

RT credentials are used to define roles which are used to represent permissions. The semantics of a given set $\mathscr{P}$ of *RT* credentials defines for each role $A.r$ the set of entities which are members of this role. The member sets of roles can also be calculated in a more convenient way using an inference system, which defines an operational semantics of *RT* language. An inference system consists of an initial set of formulae that are considered to be true, and a set of inference rules that can be used to derive new formulae from the known ones.

Let $\mathscr{P}$ be a given set of *RT* credentials. The application of inference rules of the inference system will create new credentials, derived from credentials of the set $\mathscr{P}$. A derived credential c will be denoted using a formula $\mathscr{P} \succ c$ which should be read: credential c can be derived from a set of credentials $\mathscr{P}$.

Definition 1: The initial set of formulae of an inference system over a set $\mathscr{P}$ of *RT* credentials are all the formulae: $c \in \mathscr{P}$ for each credential c in $\mathscr{P}$. The inference rules of the system are the following:

$$\frac{c \in \mathscr{P}}{\mathscr{P} \succ c} \quad (W_1)$$

$$\frac{\mathscr{P} \succ A.r \leftarrow B.s \qquad \mathscr{P} \succ B.s \leftarrow X}{\mathscr{P} \succ A.r \leftarrow X} \quad (W_2)$$

$$\frac{\mathscr{P} \succ A.r \leftarrow B.s.t \qquad \mathscr{P} \succ B.s \leftarrow C \qquad \mathscr{P} \succ C.t \leftarrow X}{\mathscr{P} \succ A.r \leftarrow X} \quad (W_3)$$

$$\frac{\mathscr{P} \succ A.r \leftarrow B.s \cap C.t \qquad \mathscr{P} \succ B.s \leftarrow X \qquad \mathscr{P} \succ C.t \leftarrow X}{\mathscr{P} \succ A.r \leftarrow X} \quad (W_4)$$

$$\frac{\mathscr{P} \succ A.r \leftarrow B.s \odot C.t \qquad \mathscr{P} \succ B.s \leftarrow X \qquad \mathscr{P} \succ C.t \leftarrow Y}{\mathscr{P} \succ A.r \leftarrow X \cup Y} \quad (W_5)$$

$$\frac{\mathscr{P} \succ A.r \leftarrow B.s \otimes C.t \qquad \mathscr{P} \succ B.s \leftarrow X \qquad \mathscr{P} \succ C.t \leftarrow Y \qquad X \cap Y = \phi}{\mathscr{P} \succ A.r \leftarrow X \cup Y} \quad (W_6)$$

There could be a number of inference systems defined over a given language. To be useful for practical purposes, an inference system must exhibit two properties. First, it should be sound, which means that the inference rules could derive only formulae that are valid with respect to the semantics of the language. Second, it should be complete, which means that each formula which is valid according to the semantics should be derivable in the system. Both properties have been shown in [45], proving that the inference system provides an alternative way of presenting the semantics of *RT* languages.

4. Time Validity in *RT*

Inference rules with time validity for RT_0 were originally introduced in a slightly different way in [40]. In this paper, we will show the extention of other languages, up to RT^T (by putting time validity constraints into this language). In this case credentials are given to entities just for some fixed period of time. It is quite natural to assume that permissions are given just for a fixed period of time, not forever.

The ability to infer credentials with incomplete information is a significant advantage of Role-based Trust management in distributed systems. However, practical applications are limited by the fact that in real life permissions can rarely be given forever. The need to revoke a credential may not be frequent, but when it occurs, it is crucial. Unfortunately, revocation of credentials is not a simple extention to the method – the system becomes non-monotonous. In

this case access rights cannot be correctly inferred without complete information about credentials or at least knowledge which credentials have been explicitly revoked, and which should be invalidated as inferred from the revoked ones. Effectively this ruins the system's scalability.

A complete solution of the credential revocation problem is beyond the scope of this paper. However, it can be partially addressed by limiting the validity of credentials to fixed periods of time. As will be shown, this does not affect the system's ability to work with incomplete information and limits the potential impact of credentials that would otherwise be revoked. An additional effect is the ability to automatically identify outdated credentials, avoiding the problem of unlimited growth of the credential database.

The restricted validity of credentials can also be used to create a system enabling certificate revocation with an arbitrary, but non-zero reaction time. Credentials valid for long periods of time would not be used directly in this case – instead they would be used to create periodically (or on request) new credentials with short validity periods. Revocation of a credential would then be a local action, no more short-term credentials would be created and the revocation would be guaranteed to be effective as soon as the last short-term credential becomes invalid.

Time dependent credentials take the form: c **in** v, meaning "the credential c is available during the time v". Finite sets of time dependent credentials are denoted by $\mathscr{C}\mathscr{P}$ and the new language is denoted as RT^T_+ (as an extention of the most powerful RT^T language) To make notation lighter we write c to denote "c **in** $(-\infty, +\infty)$". This type of time constraints can satisfy the need of negation in non-monotonic systems.

Time validity can be denoted as follows: $[\tau_1,\tau_2]$; $[\tau_1,\tau_2)$; $(\tau_1,\tau_2]$; (τ_1,τ_2); $(-\infty,\tau]$; $(-\infty,\tau)$; $[\tau,+\infty)$; $(\tau,+\infty)$; $(-\infty,+\infty)$; $v_1 \cup v_2$; $v_1 \cap v_2$; $v_1 \backslash v_2$ and v_1, v_2 of any form in this list, with τ ranging over time constants.

Time dependant credentials in wireless sensor networks can be used in a form of credential templates. Credential templates know the precise time validity of credentials and specific credentials know about narrowed period of time (for example one day). When sensor need to use a credential, it does not have to ask each time (what consume some resources) about the validity of credentials.

4.1. Inference System over RT^T_+ Credentials

Now, we can adapt inference system over RT credentials to take time validity into account. Let $\mathscr{C}\mathscr{P}$ be a given set of RT^T_+ credentials. The application of inference rules of the inference system will create new credentials, derived from credentials of the set $\mathscr{C}\mathscr{P}$. A derived credential c valid in time τ will be denoted using a formula $\mathscr{C}\mathscr{P} \succ_\tau c$, which should be read: credential c can be derived from a set of credentials $\mathscr{C}\mathscr{P}$ during the time τ.

Definition 2: from [46] The initial set of formulae of an inference system over a set $\mathscr{C}\mathscr{P}$ of RT^T_+ credentials are all in the form: c **in** $v \in \mathscr{C}\mathscr{P}$ for each credential c valid in time v in $\mathscr{C}\mathscr{P}$. The inference rules of the system are the following:

$$\frac{c \textbf{ in } v \in \mathscr{C}\mathscr{P} \qquad \tau \in v}{\mathscr{C}\mathscr{P} \succ_\tau c} \tag{CW_1}$$

$$\frac{\mathscr{C}\mathscr{P} \succ_\tau A.r \leftarrow B.s \qquad \mathscr{C}\mathscr{P} \succ_\tau B.s \leftarrow X}{\mathscr{C}\mathscr{P} \succ_\tau A.r \leftarrow X} \tag{CW_2}$$

$$\frac{\mathscr{C}\mathscr{P} \succ_\tau A.r \leftarrow B.s.t \qquad \mathscr{C}\mathscr{P} \succ_\tau B.s \leftarrow C \qquad \mathscr{C}\mathscr{P} \succ_\tau C.t \leftarrow X}{\mathscr{C}\mathscr{P} \succ_\tau A.r \leftarrow X} \tag{CW_3}$$

$$\frac{\mathscr{C}\mathscr{P} \succ_\tau A.r \leftarrow B.s \cap C.t \qquad \mathscr{C}\mathscr{P} \succ_\tau B.s \leftarrow X \qquad \mathscr{C}\mathscr{P} \succ_\tau C.t \leftarrow X}{\mathscr{C}\mathscr{P} \succ_\tau A.r \leftarrow X} \tag{CW_4}$$

$$\frac{\mathscr{C}\mathscr{P} \succ_\tau A.r \leftarrow B.s \odot C.t \qquad \mathscr{C}\mathscr{P} \succ_\tau B.s \leftarrow X \qquad \mathscr{C}\mathscr{P} \succ_\tau C.t \leftarrow Y}{\mathscr{C}\mathscr{P} \succ_\tau A.r \leftarrow X \cup Y} \tag{CW_5}$$

$$\frac{\mathscr{C}\mathscr{P} \succ_\tau A.r \leftarrow B.s \otimes C.t \qquad \mathscr{C}\mathscr{P} \succ_\tau B.s \leftarrow X \qquad \mathscr{C}\mathscr{P} \succ_\tau C.t \leftarrow Y \qquad X \cap Y = \phi}{\mathscr{C}\mathscr{P} \succ_\tau A.r \leftarrow X \cup Y} \tag{CW_6}$$

4.2. Inferring Time Validity of Credentials

This inference system evaluates maximal time validity when it is possible to derive the credential c from $\mathscr{C}\mathscr{P}$. It enhances formula $\mathscr{C}\mathscr{P} \succ_\tau c$ to $\mathscr{C}\mathscr{P} \succ\succ_v c$, specifying that at any time $\tau \in v$ in which $\mathscr{C}\mathscr{P}$ has a semantics, it is possible to infer the credential c from $\mathscr{C}\mathscr{P}$. To make notation lighter we write $\succ\succ$ to denote $\succ\succ_{(-\infty,+\infty)}$. The inference rules of the system are the following:

$$\frac{c \textbf{ in } v \in \mathscr{C}\mathscr{P}}{\mathscr{C}\mathscr{P} \succ\succ_v c} \tag{CWP_1}$$

$$\frac{\mathscr{C}\mathscr{P} \succ\succ_{v_1} A.r \leftarrow B.s \qquad \mathscr{C}\mathscr{P} \succ\succ_{v_2} B.s \leftarrow X}{\mathscr{C}\mathscr{P} \succ\succ_{v_1 \cap v_2} A.r \leftarrow X} \tag{CWP_2}$$

$$\frac{\mathscr{C}\mathscr{P} \succ\succ_{v_1} A.r \leftarrow B.s.t \qquad \mathscr{C}\mathscr{P} \succ\succ_{v_2} B.s \leftarrow C \qquad \mathscr{C}\mathscr{P} \succ\succ_{v_3} C.t \leftarrow X}{\mathscr{C}\mathscr{P} \succ\succ_{v_1 \cap v_2 \cap v_3} A.r \leftarrow X} \tag{CWP_3}$$

$$\frac{\mathscr{C}\mathscr{P} \succ\succ_{v_1} A.r \leftarrow B.s \cap C.t \qquad \mathscr{C}\mathscr{P} \succ\succ_{v_2} B.s \leftarrow X \qquad \mathscr{C}\mathscr{P} \succ\succ_{v_3} C.t \leftarrow X}{\mathscr{C}\mathscr{P} \succ\succ_{v_1 \cap v_2 \cap v_3} A.r \leftarrow X} \tag{CWP_4}$$

$$\frac{\mathscr{C}\mathscr{P} \succ\succ_{v_1} A.r \leftarrow B.s \odot C.t \qquad \mathscr{C}\mathscr{P} \succ\succ_{v_2} B.s \leftarrow X \qquad \mathscr{C}\mathscr{P} \succ\succ_{v_3} C.t \leftarrow Y}{\mathscr{C}\mathscr{P} \succ\succ_{v_1 \cap v_2 \cap v_3} A.r \leftarrow X \cup Y} \tag{CWP_5}$$

$$\frac{\mathscr{C}\mathscr{P} \succ\succ_{v_1} A.r \leftarrow B.s \otimes C.t \qquad \mathscr{C}\mathscr{P} \succ\succ_{v_2} B.s \leftarrow X \qquad \mathscr{C}\mathscr{P} \succ\succ_{v_3} C.t \leftarrow Y \qquad X \cap Y = \phi}{\mathscr{C}\mathscr{P} \succ\succ_{v_1 \cap v_2 \cap v_3} A.r \leftarrow X \cup Y} \tag{CWP_6}$$

$$\frac{\mathscr{C}\mathscr{P} \succ\succ_{v_1} c \qquad \mathscr{C}\mathscr{P} \succ\succ_{v_2} c}{\mathscr{C}\mathscr{P} \succ\succ_{v_1 \cup v_2} c} \tag{CWP_7}$$

5. RT in Wireless Sensor Networks

Sensors have a limited source of power and it is hard to replace or recharge, for example, sensors in the battle field or sensors in a large sea or forest. That is why it is so important to save these resources. On the other hand, in some cases the security of sensor network is crucial and we can use some resources to protect WSN.

Hierarchical routing, which is proposed to prolong the lifetime of WSNs, is one of the areas where it is possible to use *RT* languages. Another important area may be delegation of permissions in mobile networks, where RT^D language can be useful.

5.1. Hierarchical Routing

The hierarchical routing protocols classify sensor nodes according to their functionalities. The main purpose of such a division is to reduce the energy consumption. It is easy to delegate the privileges between nodes which are similar. The network is divided into groups (or clusters) with a leader sensor (or cluster node). The leader coordinates the activities within the group and communicates with sensors outside the own group. The different schemes for hierarchical routing mainly differ in how the leader is selected and how the sensors behave in the inter and intra-group domain.
Hierarchical routing is one of the fields where a delegation of permission from the Role-based Trust management family can be applied. For example, in a one-way communication scenario, the group leader can broadcast the message that his resources are running out, so he would like to delegate its permissions to another sensor. It can be assumed that he would do this on the condition that the potential sensor has the proper credentials. What is needed to make such decisions is information about the privileges assigned to the potential sensor by other authorities, as well as trust information about the authority itself. If the above conditions are met, the leader can delegate its permissions (and even role activation) to perform its role to another sensor which is authorized to do that.

5.2. Permissions Delegation in a Mobile Networks

Mobile sensor networks are incredibly valuable, especially in situations where traditional arrangement mechanisms fail, or are not suitable. Also, in some application scenarios such as ocean monitoring, sensors move with the ocean currents. The coverage of a mobile sensor network depends not only on the initial network congurations, but also on the mobility behavior of the sensors.
The locations covered by sensors change over time, they can regroup in order to cover the range of the new area. In this case, it is good idea to use one of the RT family languages, RT^D, which provides mechanisms to describe delegation of role activations and selective use of role membership. Sensors changing its location can delegate their permissions to other sensors. Moving from one place to another, they can change their roles and activate new ones. It is also possible to delegate some of their rights to sensors towards which they change their position. They may give up their role in favor of other sensors. They can interact with some sensors at specified periods of time, and with others in other periods of time – depending on time validity of their permissions.

6. Conclusions

Trust and trust management is an important issue in distributed wireless sensor networks. That is why it is more and more often the subject of research scientists. Because it can increase the security of the network, they can be used more widely. The concept of trust and trust management in wireless sensor network is defined in a different way, because it is used in a different cases. As it was shown above, the languages from the family of Role-based Trust management can be applied to WSN. Because of the character of this kind of network it is not suitable to use it in a small WSN where just simple low-resource wireless sensors are used, but in networks where the security is crucial. It is also possible to use RT in a wireless sensor and actuator networks.

Acknowledgements

I would like to thank Krzysztof Lasota for his contribution to this work. I would also like to thank an anonymous reviewer for very valuable and detailed comments.

References

[1] S. P. Marsh, “Formalising Trust as a Computational Concept”, Ph.D. thesis, Dept. of Computing Science and Mathematics, University of Stirling.

[2] M. Blaze, J. Feigenbaum, and J. Lacy, “Decentralized trust management”, in *Proc. 17th IEEE Symp. Secur. Priv.*, Oakland, CA, USA, 1996, pp. 164–173.

[3] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, “SPINS: Security protocols for sensor networks”, in *Proc. 7th Int. Conf. Mob. Comput. Netw. MobiCom 2001*, Rome, Italy, 2001, pp. 189–199.

[4] “Cambridge Advanced Learner’s Dictionary” [Online]. Available: http://dictionary.cambridge.org/

[5] S. Ganeriwal and M. B. Srivastava, ”Reputation-based framework for high integrity sensor networks”, in *Proc. 2nd ACM Worksh. Secur. Ad Hoc Sensor Netw.*, Washington, DC, USA, 2004, pp. 66–77.

[6] A. Boukerche, X. Li, and K. EL-Khatib, “Trust-based security for wireless ad hoc and sensor networks”, *Comp. Commun*., vol. 30, pp. 2413–2427, 2007.

[7] Z. Yao, D. Kim, and Y. Doh, “PLUS: Parameterized and localized trust management scheme for sensor networks security”, in *Proc. 3rd IEEE Int. Conf. Mob. Ad-Hoc and Sensor Syst. MASS-2006*, Vancouver, Canada, 2006, pp. 437–446.

[8] M. Gupta, P. Judge, and M. Ammar, “A reputation system for peer-to-peer networks”, in *Proc. 13th Int. Worksh. Netw. Operating Sys. Support Digit. Audio Video NOSSDAV 2003*, Monterey, CA, USA, 2003, pp. 144–15.

[9] L. Xiong and L. Liu, “Peer trust: Supporting reputation-based trust for peer-to-peer electronic communities”, *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, 2004.

[10] S. Buchegger and J. L. Boudec, “Performance analysis of the CONFIDANT protocol”, in *Proc. 3rd ACM Int. Symp. Mob. Ad Hoc Netw. Comput. MobiHoc 2002*, Lausanne, Switzerland, 2002.

[11] S. Buchegger and J. Y. L. Boudec, “A robust reputation system for peer-to-peer and mobile ad-hoc networks”, in *Proc. 2nd Worksh. Econom. Peer-to-Peer Sys. P2PEcon 2004*, Cambridge, MA, USA, 2004.

[12] P. Michiardi and R. Molva, “CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks”, in *Advanced Communications and Multimedia Security*, B. Jerman-Blazic and T. Klobucar, Eds. Kluwer 2002, pp. 107–121.

[13] Y. Rebahi, V. E. Mujica-V, and D. Sisalem, "A reputation-based trust mechanism for ad-hoc networks", in *Proc. 10th IEEE Symp. Comp. Commun. ISCC 2005*, Murcia, Spain, 2005, pp. 37–42.

[14] Y. L. Sun, W. Yu, Z. Han, and K. J. R. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks", *IEEE J. Selec. Areas in Commun.*, vol. 24, no. 2, pp. 305–317, 2006.

[15] G. Theodorakopoulos and J. S. Baras, "On trust models and trust evaluation metrics for ad hoc networks", *IEEE J. Selec. Areas in Commun.*, vol. 24, no. 2, pp. 318–328, 2006.

[16] Z. Yan, P. Zhang, and T. Virtanen, "Trust evaluation based security solutions in ad hoc networks", in *Proc. 7th Nordic Worksh. Secur. IT Syst.*, Gjovik, Norway, 2003.

[17] H. Chen, H. Wu, X. Zhou, and C. Gao, "Reputation-based trust in wireless sensor netwoks", in *Proc. Int. Conf. Multim. Ubiq. Engin. MUE 2007*, Seoul, Korea, 2007.

[18] K. Daniluk and E. Niewiadomska-Szynkiewicz, "A survey of energy efficient security architectures and protocols for wireless sensor networks", *J. Telecom. Inform. Technol.*, no. 3, pp. 64–72, 2012.

[19] K. Lasota, E. Niewiadomska-Szynkiewicz, and A. Kozakiewicz, "Adaptacja rozwiązań honeypot dla sieci czujników", in *Proc. Konferencja Sieci Komputerowe SK-12*, Szczyrk, Poland, 2012, *Studia Informatica*, vol. 33, no. 3A, pp. 139–148 (in Polish).

[20] K. Lasota, E. Niewiadomska-Szynkiewicz, and A. Kozakiewicz, "Mobilny honeypot dla sieci sensorycznych", *Przegląd Telekomunikacyjny*, no. 8–9, pp. 699–704, 2012 (in Polish).

[21] M. Momani, S. Challa, and K. Aboura, "Modelling trust in wireless sensor networks from the sensor reliability prospective", in *Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecomm.*, T. Sobh *et al*., Eds. Heidelberg: Springer, 2007, pp. 179–189, pp. 317–321.

[22] R. A. Shaikh, H. Jameel, S. Lee, S. Rajput, and Y. J. Song, "Trust management problem in distributed wireless sensor networks", in *Proc. 12th IEEE Conf. Embedd. Real-Time Comput. Syst. Appl. RTCSA 2006*, Sydney, Australia, 2006, pp. 411–414.

[23] E. Aivaloglou and S. Gritzalis, "Hybrid trust and reputation management for sensor networks", *Wirel. Netw.*, vol. 16, no. 5, pp. 1493–1510, 2010.

[24] Z. Yao, D. Kim, I. Lee, K. Kim, and J. Jang, "A security framework with trust management for sensor networks", in *Proc. Worksh. of the 1st Int. Conf. Secur. Priv. Emerg. Areas in Commun. Netw. SecureComm 2005*, Athens, Greece, 2005, pp. 190–198.

[25] Y. Zhiying, K. Daeyoung, L. Insun, K. Kiyoung, and J. Jongsoo, "A security framework with trust management for sensor networks", in *Proc. 1st Int. Conf. Secur. Priv. Emerg. Areas in Commun. Netw. SecureComm 2005*, Athens, Greece, 2005.

[26] T. A. Zia, "Reputation-based Trust Managmenet in Wireless Sensor Networks", in *Proc. Int. Conf. Intell. Sensors, Sensor Netw. Inform. Proces. ISSNIP 2008*, Sydney, Australia, pp. 163–166.

[27] M. Momani and S. Challa, "Trust management in wireless sensor networks", in *Proc. 5th ACM Conf. Embedded Netw. Sensor Syst.*, Sydney, Australia, 2007.

[28] H. Chen, H. Wu, X Zhou, and C. Gao, "Agent-based trust model in wireless sensor networks", in *Proc. 8th ACIS Int. Conf. Softw. Engin. Artif. Intell., Netw. Parallel/Distrib. Comput*., Qingdao, China, 2007.

[29] D. F. Ferraiolo, R. S. Sandhu, S. I. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control", *ACM Trans. Inf. Syst. Secur*., vol. 3, pp. 224–274, 2001.

[30] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, nad C. E. Youman, "Role-based access control models", *IEEE Computer*, vol. 2, pp. 38–47, 1996.

[31] M. Blaze, J. Feigenbaum and M. Strauss, "Compliance checking in the policymaker trust management system", in *Proc. 2nd Int. Conf. Finan. Cryptogr*., London, UK, 1998, pp. 254–274.

[32] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, "The role of trust management in distributed systems security", in *LNCS Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, J. Vitek and C. D. Jensen, Eds. Springer, 1999, pp. 185–210.

[33] D. Clarke, J.-E. Elien, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest, "Certificate chain discovery in SPKI/SDSI", *J. Comp. Secur.*, vol. 9, pp. 285–322, 2001.

[34] P. Chapin, C. Skalka, and X. S. Wang, "Authorization in trust management: features and foundations", *ACM Comp. Surv*., vol. 3, pp. 1–48, 2008.

[35] N. Li and J. Mitchell, "RT: A Role-Based Trust-Management Framework", in *Proc. 3rd DARPA Inform. Survivabil. Conf. Expos*., Washington, DC, USA, IEEE Computer Society Press, 2003, pp. 201–212.

[36] N. Li, J. Mitchell, and W. Winsborough, "Design of a role-based trust-management framework", in *Proc. IEEE Symp. Secur. Priv.*, Oakland, CA, USA, 2002, pp. 114–130.

[37] N. Li, W. Winsborough, and J. Mitchell, "Distributed credential chain discoveryin trust management", *J. Comp. Secur*., vol. 1, pp. 35–86, 2003.

[38] M. R. Czenko, S. Etalle, D. Li, and W. H. Winsborough, "An introduction to the role based trust management framework RT", Tech. Rep. TR-CTIT-07-34, Centre for Telematics and Information Technology University of Twente, Enschede, 2007.

[39] K. Lasota and A. Kozakiewicz, "Model of user access control to virtual machines based on RT – family trust management language with temporal validity constrains – practical application", *J. Telecom. Inform. Technol.*, no. 3, pp. 13–21, 2012.

[40] D. Gorla, M. Hennessy, and V. Sassone, "Inferring dynamic credentials for role-based trust management", in *Proc. 8th ACM SIGPLAN Conf. Princip. Pract. Declarat. Program. PPDP 06*, Venice, Italy, 2006, pp. 213–224.

[41] A. Felkner and A. Kozakiewicz, "RT^T_+ – time validity constraints in RT^T language", *J. Telecom. Inform. Technol.*, no. 2, pp. 74–82, 2012.

[42] W. M. Grudzewski, I. K. Hejduk, A. Sankowska, and M. Wańtuchowicz, *Trust Management in Virtual Work Environments: A Human Factors Perspective*. CRC Press, 2008.

[43] A. Felkner and K. Sacha, "The semantics of role-based trust management languages", in *Advances in Software Engineering Techniques*, T. Szmuc, M. Szpyrka, J. Zendulka, Eds., *LNCS 7054*. Heidelberg: Springer, 2012, pp. 179–189.

[44] A. Felkner and A. Kozakiewicz, "Kontrola dostępu w rozproszonych systemach – trzy semantyki języka RT^T", in *Proc. II Konferencja i3: internet – infrastruktury – innowacje*, Wrocław, Poland, 2010 (in Polish).

[45] A. Felkner and K. Sacha, "Deriving RT^T credentials for role-based trust management", *e-Inform. Software Engin. J.*, vol. 4, no. 1, pp. 9–19, 2010.

[46] A. Felkner and A. Kozakiewicz, "Time validity in role-based trust management inference system", in *Proc. Int. Worksh. Sec. Trust Comput., Data Manag., and Appl. IWCS-11*, Loutraki, Greece, 2011, *Communications in Computer and Information Science*, Springer, 2011, vol. 187, pp. 7–15.

Anna Felkner graduated from the Faculty of Computer Science of Białystok University of Technology (M.Sc., 2004) and the Faculty of Electronics and Information Technology of Warsaw University of Technology (Ph.D., 2010). At present she is an Assistant Professor at Network and Information Security Methods Team in NASK Research Division. Main scientific interests concern the security of information systems, especially access control and trust management.

E-mail: anna.felkner@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland