İSTANBUL UNIVERSITY CERRAHPAŞA

RESEARCH ARTICLE

Electrica 2020; 20(1): 86-96

A Hybrid Trust-Modeling Approach for IoT Security

Şerif Bahtiyar

Department of Computer Engineering, Security and Privacy Research Lab, İstanbul Technical University, İstanbul, Turkey

Cite this article as: Bahtiyar Ş. A Hybrid Trust-Modeling Approach for IoT Security. *Electrica*, 2020; 20(1): 86-96.

ABSTRACT

The Internet contains many types of services with different security options, which may be updated very frequently. An entity selects only those services in trusted networks that satisfy its security requirements. The primary challenge is to determine the trust level of services in such networks, which contain devices with limited processing power, such as Internet-of-Things (IoT) devices. We propose a hybrid approach to formulate the trust of a security system in IoT networks. Our approach provides a systematic way to model the trust of security from the viewpoint of an entity. Furthermore, using a case study, we evaluate the approach via simulations of a smart entity that has to select a trusted network. The evaluation results show that our approach provides satisfactory and flexible trust computations.

Keywords: Security, trust, hybrid, privacy

Corresponding Author:
Şerif Bahtiyar

E-mail:
bahtiyars@itu.edu.tr

Received: 19.12.2019

Accepted: 02.01.2020

DOI: 10.5152/electrica.2020.19090

Introduction

The prevalentusage of the Internet with low-cost computing equipment has made entities connected to services in Internet-of-Things (IoT) devices. An entity tends to interact with a service in such an environment if the entity truststhesecurity of a service that is a significant challenge to increase the benefit of interactions.

Security includes topics such as protection of information from theft and corruption. Most of the time, the goal of security is to preserve confidentiality, integrity, and availability [1]. However, trust has a subjective nature and diverse meanings [2]; therefore, thetrust perceptions of entities are different. However, thediversity of meaning is not good for security that requires clarity and precision. Therefore, trust necessitates precise technical definitions that may mislead the public. The existingtrust models are insufficient inrepresenting dynamic and subjective needs of different entities in a cyberspace, where low computing devices operate, such as IoT devices.

This study is motivatedbythe lack of a common trust model that satisfies the dynamic security needs of a specific entity for IoT networks. In addition, security solutions have been improved, and new security mechanisms have been introduced. Therefore, entities have changed their requirements accordingly. Becausetrust is subjective and highly dependent on the context, it is impossible to have a common trust model for alltheentities in different IoT networks. Therefore, we need an approach to use multiple models simultaneously in order to represent and compute the trust of entities in relation tothesecurity.

In this study, we extend anhybrid approach, namely, Core–Crust Modeling Approach (CCMA), in [3] to model trust in relation to security according tothedynamic requirements of a specific entity for IoT networks. The approach combines many trust models to satisfythedynamic needs of each entity. The main contributions of our research are twofold as follows.

- First, we introduce a novel computation approach for trust modeling that representsthespecific security requirements of an entity from networks.
- Second, we show how to integrate different models according to CCMA to satisfythetrust-computation requirements of a specific entity.

The rest of this study is organized as follows. We examine the trust and security modeling issues in Section II. We present our approach in the next section. Section IV discusses the evaluation and applicability of the proposed approach. This study is concluded in Section V.

Security and Trust Research

Trust is widely used for decision making, anditis defined differently in information systems [4]. For instance, in the context of access control, trust is defined as the possession of authentic and valid credentials necessary for access control at an end point [5]. To have a concrete trust model, trust should be defined precisely. Here, we follow the trust definition provided in [6], where *Trust is the security expectation of an entity from a service according to the available security-evaluation information of that entity.*

In this study, theavailable security information is related to security from the viewpoint of an entity. Because theavailable security information depends on entities, each entity may have different security-evaluation information related to the security of a specific service.

Properties of Trust Relationships

The propertiesof a trust relationship determine the definition of trust in a particular context. Two entities may have one-to-one trust relationships, which may be an asymmetric relationship. A trust relationship may be one-to-many, many-to-many, or many-to-one. Specifically, a trust relationship depends on the entities involved in that trust relationship [7]. If trust information is delegated from one entity to another in a group, the model uses the transferable property of trust among entities that is applied in the group [8].

Subjectivity, which is another property of trust, depends on the personal opinions of entities. Mostly, like security, trust is always used in a particular context; therefore, trust is highly context dependent. Generally, trust values are used to represent different degrees of trust related to an entity, and the values represent the measurable properties of trust.

The dynamicnature of trust is another significant property. This means that trust may change according to different factors, such as time, actions of entities, and events in the environment. However, thedynamic property of trust complicates its modeling, which should be, therefore, performed using adaptive approaches such asartificial-intelligence algorithms [9].

General Trust Models

Trust models determine the degree of a trust relationship between two entities. Trust modeling is defined as *the process of defining a complementary threat profile for a security architecture based on an acceptable trust model* [10]. Trust modeling must include implicit or explicit validation of an entity's identity or characteristics. These facts are necessary for a particular event to occur. However, an entity may have lack of information regarding the identities of other entities it interacts with. In this case, a trust model should conter the aforementioned lack of information. Thus, there are following three types trust models: direct, transitive, and assumptive.

In the direct trust model, the credentials of an entity are validated without depending on any other entity [10]. In addition, trust is established via observations, and there is no propagation of trust over entities in this model. The advantage of direct trust model is that the validation of credentials is realized by the same entity. Therefore, a high level of confidence is ensured in every entity. However, the disadvantage of the model is that it may be more labor intensive and more expensive than other models. Public Key Infrastructure (PKI) is an example for the direct trust model.

In the transitive trust model, trust is transmitted through other parties [11]. This model is also known as the indirect trust model [8]. It is based on the delegation of trust from one entity to anotherone. For instance, entity *A* has direct trust to entity *B*, and entity *B* has direct trust to entity *C*;therefore, entity *A* has transitive trust to entity *C*. However, entity *A* does not need to validate trust to entity *C*.

However, in some situations, trust is not transitive [12]. Assumptive trust model is the formal name of spontaneous trust, and it does not necessitate any validation process. The pretty-good-privacy web of trust is a kind of this trust model. The difference between the transitive trust model and the assumptive trust model is the validation process.

Trust Management

Trust is considered a complementary solution to the traditional security solutions to ensure high security. Traditional decentralized trust-management architectures do not directly address questions, such as policy changes under rapidly changing network conditions. For instance, achieving fine-grained access control is possible by using a formal specification with a policy language [13]. Trust and risk may be used in role-based access control policies in decentralized networks [14]. An automatic network-security approach, which is based on multi-dimensional trustworthiness, is proposed in [15].

Recently, there have been significant research efforts on trust related to thes ecurity of services. For instance, the number of trust models has increased for information systems; however, there is a lack of interoperability among these models; therefore these models are not easily applicable [16] and [17]. Another example is security as a service [18], where trust models are used.

Cryptography has become a significant tool for establishing trust in IoT networks. For instance, PKI-based trust models are cryptographic trust solutions that use digital certificates, such

as X.509 certificates [19]. However, complete cryptographic information is needed to compute trust metrics by usingthese models. PKI- or X.509-certificate-based trust models do not always represent the subjective needs of entities.

In the literature, different cryptographic methods have been used to establish trust. For instance, some trust models consider the identity information of entities and services to compute trust metrics, where identity management is a significant challenge. Identity-based cryptology is used to cope with -the identity management challenge [20]. In addition, zero-knowledge proof is also used to bind the identities of entities in some trust models [21]. However, Trusted Computing Group (TCG) uses an open-standard-based interoperability framework to defend against attacks on both hardware and software implementations. Particularly, the hardware specified by TCG is called the Trusted Platform Module (TPM),which uses cryptographic operations to establish the root of trust. TPM can store some cryptographic keys to perform cryptographic operations [22]. The strength of a cryptographic solution may determine the level of trust.

Trust is a soft security mechanism that is used with some other mechanisms such as recommendation and reputation in order to establish security goals, such as privacy in IoT networks. Hard security mechanisms such as cryptographic solutions are inadequate to ensure privacy in online social networks and platforms. For example, preserving the privacy of identities in virtual marketing campaigns in social networks is a challenging task, which is impossible to accomplish using only hard security mechanisms [23].

Mostly, software agents represent entities in the cyberspace, where they interact with many IoT devices. For instance, the k-Nearest Neighbor (kNN) query under the preference of a user is used for privacy-preserving purpose [24], which may be applied on IoT devices in electronic - commerce (e-commerce). There are many soft security solutions to preserve privacy in e-commerce. The trust and risk perceptions of users affec tthe market success of the products. A mediation model in [25] uses trust and risk to increase the success of product by preserving customer privacy. A more recent research regarding IoT and trust uses blockchain to secure industrial IoT applications, where there is a lack of trust in the environment [26]. In summary, the existing trust models do not satisfy all the requirements of a particular entity because there is no interoperability among trust models in the cyberspace.

Hybrid Trust-Modeling Approach for Security

Trust is subjective and entities should have their own trust models that are dynamic. We introduce a trust-modeling approach related to security; the approach is convenient for application in IoT networks. The approach is entity centric and it is used to represent trust on the basis of the needs of a particular entity about the security of a service in IoT networks. Specifically, an entity can handle dynamic-security properties according to its own needs in order to compute trust related tothe security of a service.

Security Properties from the Viewpoint of an Entity

We represent the security properties of a service by using a set of atomic units, as given in [6]. A set with n elements, $\Phi_x = \{\phi_1,...,\phi_n\}$, represents the atomic unit set of service x in a specific entity, where ϕ represents an atomic unit. In our approach, each entity may have different security representations from the security system of a service. Therefore, entities represent a security system according to their own needs. In this study, a security system is a set of security mechanisms. The security system is responsible to accomplish security goals, such as preserving the privacy of interacting entities with the service.

An *atomic unit* can be a property of a security mechanism. The atomic unit may also be a set of some properties. Moreover, an atomic unit may be a security mechanism or a set of security mechanisms. For example, assume that a security system has Data Encryption Standard (DES) and Advanced Encryption Standard (AES) encryption algorithms for the encryption of network traffic; these algorithms are the security mechanisms used in our approach. Additionally, assume that 128, 192, and 256 key-size options are available for AES encryption in the security system. An entity may represent both DES and AES with one atomic unit like $\Phi = \{\phi_{enc}\}$. However, another entity may distinguish DES and AES; therefore, the entity may represent both DES and AES by using different atomic units, $\Phi = \{\phi_{des}, \phi_{aes}\}$. Furthermore, the key size may be a significant property for another entity; therefore, the entity may have different atomic-unit representations for three key-size options, $\Phi = \{\phi_{des}, \phi_{aes128}, \phi_{aes192}, \phi_{aes256}\}$.

Security mechanisms are dynamic; therefore, the set representation of a security system depends on time. This purpose of a security mechanism has to be considered from the viewpoint of an entity. For instance, the security system of a service c in an entity is represented usinga set of atomic units as follows:

$$\Phi_c(t) = \{\phi_1,...,\phi_n\}$$

The resource limitations of entities result in them having different granularity about the representation of a security system. The trust models of each entity provide better computation results with different granular representations of security systems because an entity uses granularity to better represent its security and privacy needs from the security system of a service. For example, symmetric encryption mechanisms may perform better than asymmetric ones for a resource-limited entity, such as an IoT device. Therefore, in this case, each encryption mechanism has to be represented using an atomic unit. However, another entity may have no resource limitation; therefor, it may use any encryption mechanism, and all the mechanisms may be represented usingone atomic unit as that in the above-mentioned example.

Hybrid Approach for Trust Modeling

We present a top–down approach for representing and modeling of trust related to the security system of a service according to needs of an entity. Our approach results in a hybrid model-

ing approach that contains two types of models, namely, a core model and many crust models. We refer to such a modeling approach as CCMA as in [3]. CCMA is outlined in Figure 1.

In the proposed approach, entities should have a common trust model. We refer to a core model of an entity that represents the common model. In Figure 1, CORE represents the core model of an entity. The entitiesare expected to have the same core model, which may not be modified frequently. The core model represents the common requirements of entities from security systems. It also enables the entities to interact and communicate with each other in the cyberspace. We observe that entities must interact with many other entities and services to gather information for more accurate trust computations from many networks. For example, the framework in [27] may be used as a core model for entities in virtual organizations. In CCMA, CORE enables entities to establish security interoperability with other entities and services during interactions.

Trust is a subjective property, and we represent it by using CRUST models in CCMA. The CRUST of an entity may contain many models about the security system of a service. Moreover, each entity may have different crust models about the security system of the service. A crust model may be modified very frequently depending on the recent needs of the entity. Crust models are used to represent the dynamic and specific needs of an entity from services. Because new security attacks have occurred and new solutions have been introduced, the security needs of entities have changed too. Therefore, trust models have to counter with these changes dynamically that means trust models have to be adaptive. Artificial-intelligence algorithms are the best candidates to provide the best integration options for the trust-computation models of CRUST.

In IoT networks, entities may have various crust models simultaneously that may be changed very frequently according to the recent needs of entities. For instance, an entity may use an identity-based approach to enhance its security and privacy, where trust may be modeled usinga crust model for Peer to Peer (P2P) systems [28]. Another entity may need to use zero-knowledge proof based trust model for authentication, e.g., crust model such as Pseudo Trust [29]. However, crust models may also change when the entity moves from one context to another. For example, if an entity moves from the e-commerce context to electronic-health (e-heath) one, it may choose a different crust model [30] to adapt to the e-health context.

A crust model is connected to the core model with at least one parameter. Particularly, a crust model depends on the core model; therefore, the relation between the core model and the crust model is ensured by parameters of both models. The crust model may be designed to determine a parameter of the core model. A connection between the core model and the crust model is represented by an intersection of the core model and the crust model. For instance, we represent the connection between *CORE* and crust model *C*1 with the intersection of *CORE* and *C*1 in Figure 2. The intersection between the crust model and the core model represents the relationship between both the models.

In CCMA, no trust relationship exists between models. Therefore, the intersection of a core model and crust models or intersections between crust models do not represent any trust relationship. Core model, crust models, and intersections create the overall trust model in an entity. For instance, the trust-management model in mobile adhoc networks [31] may be used as the core model in an entity. The information flowmodel in [32] may be used as a crust model. Here, the crust model and the

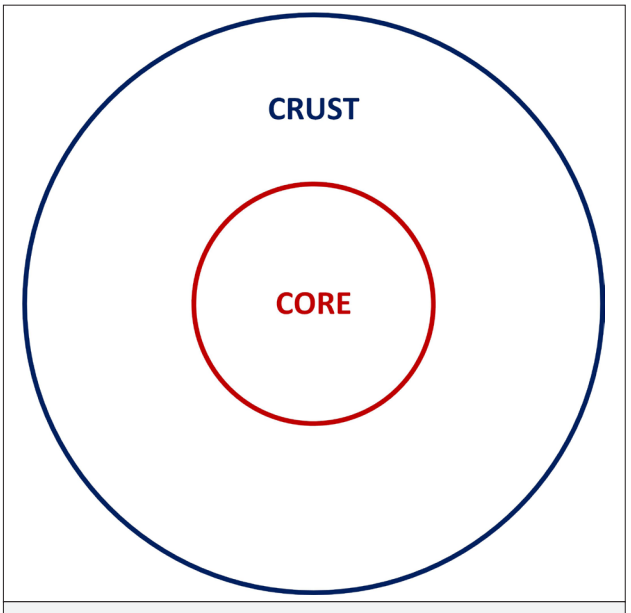


Figure 1. Core–crust modeling approachof a security systeminanIoT network

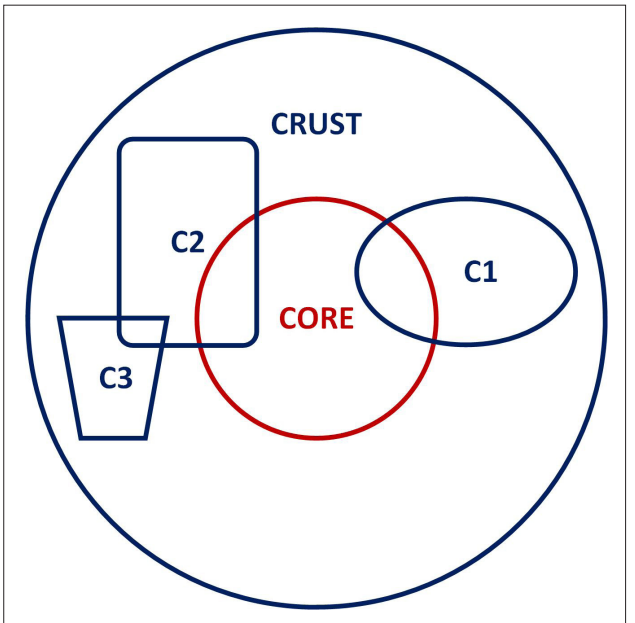


Figure 2. Correct example for CCMA

core model share the security and privacy information parameters for trust computations. The crust model assists the core model to achieve more accurate trust-computation results.

In the proposed approach, common parameters among the models must be estimated. If all the models are designed by considering possible coexistence, the entities will have better trust-computation solutions. A more specific crust model may be used to determine the parameter of another crust model more precisely. In this case, the crust model and the core model may not share any parameter. However, the crust model has to be connected with the core model indirectly over another crust model. For example, crust model *C*3 is connected to the core model over crust model *C*2, as depicted in Figure 2. Crust models *C*1 and *C*2 are connected directly to the core model.

Consider the relationship between the core model and the crust model in the previous example, where the crust model is directly connected to the core model and is used to obtain information for trust computations by the core model as a real example. Assume that the crust needs a model to obtain more specific security-evaluation information for trust computations. In addition, assume that the entity uses the model presented in [6] as a crust model to assist the existing crust model to obtain more information. Now, the entity has one core model and two crust models, where only one crust model is connected directly to the core model, and the other crust model is connected indirectly, such as crust models *C*2 and *C*3 in Figure 2.

All the crust models are connected to the core model. For example, crust model *C*1 is independent of the core model, as depicted in Figure 3, which is an incorrect example. CCMA ensures an entity to have its own trust metrics. An entity may compute metrics on the basis of its own needs by using many models. The essential requirement for many models is that they have to share some parameters. Specifically, if a model does not contribute to the trust computations in the core model, it is not a part of CCMA. From that viewpoint, all the crust models in an entity must share a parameter with the core model directly or indirectly. The motivation of this approach is that an entity should integrate its static part with its dynamic part. For instance, the two crust models in the example provided in the previous paragraph are connected to the core model.

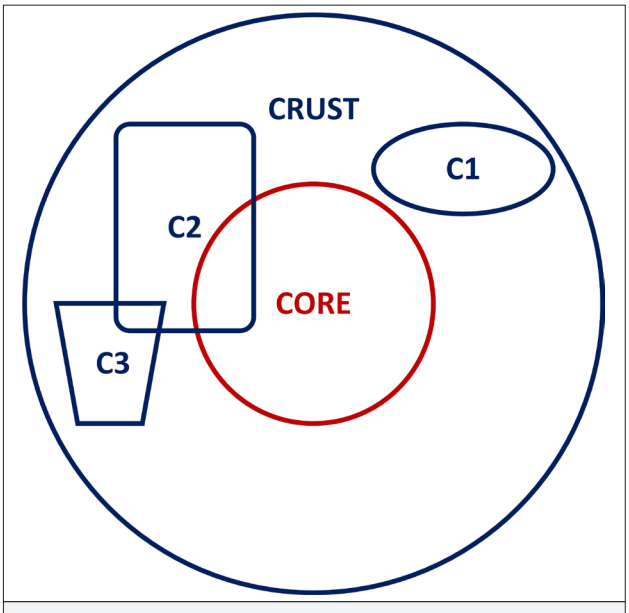


Figure 3. Incorrect example for CCMA

An entity may very frequently change the crust models that represent the dynamic part of CCMA. However, assume that the entity intends to use a new algorithm for trust computations in order to improve its existing trust solution, such as SUNNY algorithm [33],which does not share any parameter with the core model and existing crust models. Because SUNNY shares no common parameters with other models, the entity cannot use the algorithm for its trust computations without some modifications in the existing structure of models, preferably SUNNY.

Some existing entity-centric trust models can be improved and applied to many entities for trust computations by using CCMA in IoT networks. Therefore, our trust-modeling approach is dynamic and open to improvement.

Applicability and Evaluation

In this section, we show some potential applications of the hybrid approach to explain its usage. We also compare CCMA with the existing trust-computation models to present the differences between CCMA and other models. Furthermore, we have explained CCMA by usinga case study. The case study aims to show the applicability of CCMA by using numerical examples over a realistic scenario. We simulated the case study via MATLAB 2011a, following which we presented the results for a smart entity running on IoT networks.

Some Potential Applications of Hybrid Approach

CCMA integrates trust models to provide effective trust computations for entities. In this section, we show two potential applications of CCMA.

Recently, online shopping has grown rapidly, where IoT devices have become apparent. Additionally, security and privacy challenges have also increased considerably because of various services interacting with low-processing IoT devices. For instance, almost every day, we read or watch stories from magazines, newspapers, or televisions regarding the security breaches of online shops, often resulting in financial loss for their customers. The primary reason for the loss is weak security systems, which decrease trust in online shopping. Although there are many security solutions for online shopping, there is no solution to establish trust regarding these security solutions on the basis of individual needs of customers.

A customer may use CCMA to compute the trust about the security system of online-shopping systems that are based on security and privacy needs of a customer. For instance, assume that two customers need to buy some electronic equipment from online shops. They have entities that represent the

customers. In addition, assume that the security needs of the customers differ from each other, thereby requiring different trust-assessment solutions. Specifically, customer ϵ_1 considers cryptographic algorithms in the TPM of servers owned by online shops as the security-evaluation information to compute trust. Furthermore, customer ϵ_1 computes its trust metric usingcore model CO. ϵ_1 also uses crust model *CR*1 to obtain the security-evaluation information. *CR*1 extracts the security-evaluation information directly by observing the TPMs. Additionally, customer ϵ_1 can provide the recommendations about security systems by using another crust model *CR*2;however, it does not use any recommendation for trust computations. However, customer ϵ_2 uses *CO* as the core model and *CR*2 as thecrust model. Contrary to ϵ_1, ϵ_2 computes trust metrics only by using recommendations, and it receives recommendations by using *CR*2. This example shows that each entity may have different models to compute trust metrics according to its own needs by using CCMA.

Hospitals contain many IoT devices, and they provide Internet access to some of the services that may contain private information about patients, such as patients' personal registration data and blood-sample results. The patients may care about their privacy when they apply to hospitals, as the private information of patients may be revealed. This may result in financial loss for patients. For instance, revealing the private information of patients may increase the health-insurance costs of patients, or it may be ethically uncomfortable condition. Therefore, the patients need to trust the security of hospitals' information systems. However, each patient has different security requirements; therefore, they may compute trust about a security system differently. Moreover, patients' trust-computation models may change depending on their conditions and security threats. In this case, the entities that represent the patients may use CCMA for trust computations. For example, assume that patient *PAT* will have a new baby after eight months. In addition, assume that the patient recently changed her job and that her current circumstance may prevent her from having a better job allocation. This circumstance prevents her from revealing this fact before three months of birth. However, she needs examination; therefore, she has to choose one of the six hospitals in her city and must fix an appointment for the examination.

Each hospital has different security systems of their information systems. Therefore, *PAT* must compute the trust of each hospital. She has core model *COH* for trust computations. *COH* brings security and privacy information from each hospital and, subsequently, computes the trust using this information. However, the current case is more sensitive; therefore, she must update her trust model temporarily. She also needs strong cryptography based access control for patients' data to protect her private information. However, her entity is unable to test all the cryptographic security properties of a security system; therefore, she needs recommendations from some of her friends. Therefore, she has updated her trust-computation solution to be able to use the recommendations. Specifically, she integrates recommendation model *REC* to its trust-computation solution. *REC* is responsible to obtain cryptographic information from her friends, and she feeds *COH* with such information. In this case, *REC* is her crust model, and it is integrated with her trust-computation solution. However, it may be removed after many months depending on her needs. This example shows how dynamic needs of an entity may be usedina real application by using CCMA.

The subjective nature of trust and the need for adaptive trust solutions motivate us to use CCMA. The above-mentioned examples show how CCMA can be usedin real-life applications. We believe that adaptive trust solutions that integrate the existing trust models with new trust models will be strong candidates for emerging trust-computation solutions in IoT networks.

Comparison of Trust-Computation Models

Trust modes have been designed according to their contexts, information-gathering approaches, and some subjective factors. The context usually determines the trust relationships among entities. For instance, entities in social networks and IoT networks may have different trust-computation models. In social networks, entities are generally people that interact with one another, whereas in IoT networks, software agents or sensors are usually entities. In our approach, trust models are used for security context in the cyberspace.

The amount of information is a significant issue to achieve accurate trust-computation results. Information may propagate directly from one entity to another, indirectly over many entities, or both directly and indirectly. For instance, the statistical trust establishment in wireless sensor networks [34] uses a direct information flow model, whereas an indirect information flow model is used for trust management in mobile ad hoc networks [31]. However, the information-theoretic trust-modeling approach [35] considers both direct and indirect information for trust computations. In CCMA, an entity may use either direct or indirect information for trust computations. These properties of the context and needs of an entity determine which models may be used to compute trust metrics among a large number of different computation models.

The existing trust models are generally static models, which are inappropriate for dynamic contexts. For example, the changing properties of communication systems, such as updating algorithms of a cognitive radio solution, may be inappropriate to be represented using static trust models. We classify such trust models as static models. However, computer technology has been changing rapidly and trust models should be adaptive to counter these changes. Therefore, we need adaptive models. CCMA can provide adaptive trust models for security context.

Mostly, subjective factors represent the specific needs of entities. The number of these factors in a trust-computation model determines how the model represents the needs of entities. The existing trust-computation models have different number of subjective factors. For example, the trust-computation mod-

Table 1. Comparison of trust models

Approach	Context	Modeling	NSF	Advantages	Disadvantages
Guha, 2004 [32]	Social networks	Transitive Static	3	1) Distinguishes trust and distrust 2) Has a formal framework	1) May not be suitable for small-scale networks 2) Does not consider security
Gray, 2003 [36]	Security within small world networks	Hybrid Static	2	Simple	1) Fake recommendations 2) Problem with multiple short paths 3) Does not consider cryptographic security solutions
Velloso, 2010 [31]	Ad hoc networks	Transitive Static	6	Needs less processing power and less memory	1) Nodes should monitor neighbors all the time to construct and update trust relations 2) Does not consider security
Probst, 2007 [34]	Sensor networks	Direct Static	4	1) Considers history for trust computations 2) No single point of failure	1) Computationally complex to determine the t-distributions 2) Does not consider security
Sun, 2006 [35]	Ad hoc networks	Hybrid Static	1	1) Generic: can be applied to many networks 2) Contains two models	1) Requires additional hardware to sense the neighbors 2) Does not consider cryptographic security solutions
CCMA	Security	Hybrid Dynamic	Unlimited	1) Security specific: considers cryptographic security solutions 2) Adaptive	1) Models have to share some parameters 2) May need formal models for computing subjective factors 3) Security specific

el in [35] has one subjective factor, whereas the one in [31] has six subjective factors related to ad hoc networks. The number of factors in an entity may change time to time; therefore, a constant number of subjective factors may be inadequate to represent the dynamic needs of entities. In CCMA, an entity may have different number of subjective factors to adjust dynamic conditions.

A detailed comparison among different trust-modeling approaches with respect to context, information model, number of subjective factors (NSF), advantages, and disadvantages is provided in Table 1. Although CCMA is specific to security and privacy context, it may be easily adapted to some other contexts. Our approach is dynamic and it may have different NSF.

Case Study: A Smart Entity for Network Selection

This case study aims to show the applicability of CCMA. In the case study, a user has a smart phone that is able to change its communication network to access the Internet when the user moves. The smart phone changes the network according to the decisions of a smart entity running on it. The smart entity is responsible to assess and select a trusted communication network by using CCMA in a specific location.

Assume that the user goes to an airport. The IT department of the airport provides free Internet access to passengers via five different Wireless Local Area Networks (WLAN)s. However, the user has to pay for each bit for its current Internet connection. Therefore, the smart entity assesses the trust of WLANs, and, subsequently, it connects to more trusted WLAN. In addition, WLANs contain many services with different security properties. There are also various passengers connected to WLANs with different intentions and IoT devices that may cause many security and privacy threats for users.

Table 2. Security-evaluation information ranges received by the smart entity from communication networks

	Experience		Recommendation	
Network	**Min**	**Max**	**Min**	**Max**
Initial	0.5	0.8	0.4	0.6
Net1	0.6	0.9	0	0
Net2	0.5	0.6	0	0
Net3	0	0.3	0	0
Net4	0.8	1	0	0
Net5	0.6	0.8	0	0

The smart entity computes trust using a core model, namely TM. The core model uses the experiences and recommendations about the security system of communication networks to compute trust. Initially, the smart entity has a crust model, EM, to obtain experiences and a crust model, TM, to receive recommendations. The relationships among models are depictedin Figure 4. When the user moves to the airport, the smart entity updates its trust-computation model by using the available

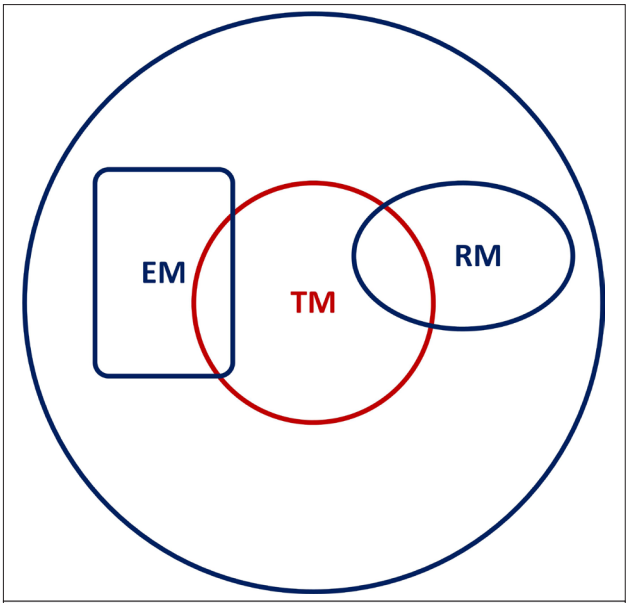


Figure 4. Trust computations usingexperience and recommendation crust models

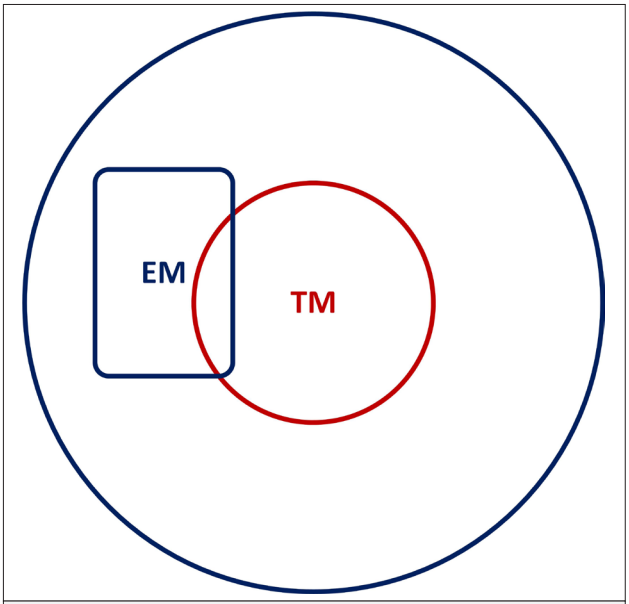


Figure 5. Trust computations using only experience crust model

security-evaluation information in the airport. Specifically, the smart entity has no recommendation with high confidence, and it does not use recommendations to compute trust. In this case, the smart entity computes trust usingTM and EM only. In the airport, the trust-computation model of the smart entity is depicted in Figure 5.

An entity may compute trust of each atomic unit in $\Phi(t)$ by using CCMA. The trust of an atomic unit is computed using Equation 1 in TM. For the sake of brevity, security systems of services are represented using one atomic unit, and they do not

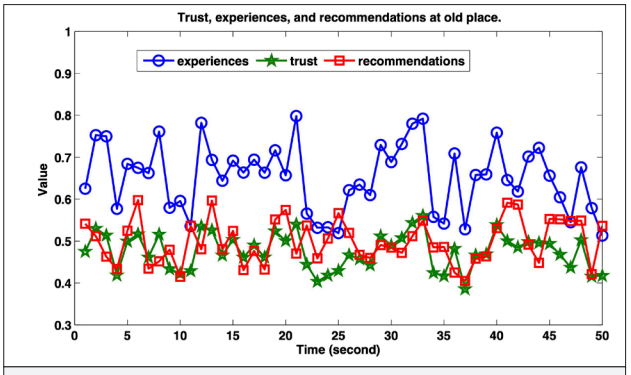


Figure 6. Trust and security-evaluation information before the smart entity moves

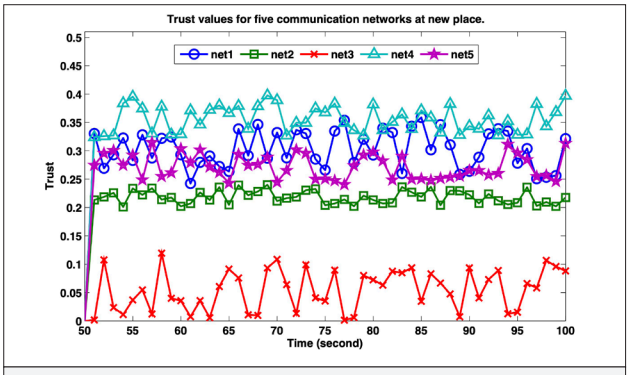


Figure 7. Trust information after the smart entity moves

change with time, such that $\Phi = \{\phi\}$. In this case study, the security-evaluation information is uniformly distributed between the ranges shown in Table 2. in all communication networks. One has the following:

$$T(t) = \alpha(t)E(t) + \beta(t)R(t),\ 0 \le \alpha(t) + \beta(t) \le 1. \quad (1)$$

where

$$0 \le \alpha(t), \beta(t), E(t), R(t) \le 1$$

Initially, the smart entity computes trust using experiences and recommendations, where $\alpha(t) = 0.5$ and $\beta(t) = 0.3$. Experiences, recommendations, and trust values are shown for the communication network for old location in Figure 6. In this model, the maximum trust is always below the maximum values of either experience or recommendation in a specific time, as depicted in Figure 6. When the user moves to the airport at time=51, the smart entity updates its trust-computation model, where $\alpha(t) = 0.4$ and $\beta(t) = 0$. The entity computes the trust of five WLANs, as depicted in Figure 7. Because all the WLANs are free and Net4 is generally more trusted than others, the smart entity selects Net4 to access the Internet.

The case study shows the applicability of the proposed entity-centric approach regarding trust modeling. Additionally, the numerical results in the case study show how trust-based decisions may be made. It is obvious that dynamic trust models are

needed and that CCMA provides adaptive trust-computation solutions.

Conclusion

Because IoT networks are a part of the cyberspace, achieving trust computations using limited processing power has become a significant challenge for entities in order to select the best possible service. In this study, we presented a hybrid, trust-modeling approach based on the requirements of entities. The proposed approach, CCMA, is convenient for IoT networks. The core model represents the common requirements of entities from security systems. The crust model represents the specific needs of an entity. In CCMA, an entity may have more than one crust model, but it has only one core model. The hybrid approach is suitable for satisfying subjective and dynamic needs of different entities related to security and privacy services on IoT networks.

We analyzed CCMA by using comparisons and empirical evaluations. Specifically, we compared CCMA with the existing trust models and showed the applicability of the proposed approach. We also presented a case study using a smart entity for performing network selection. The results of the case study showed that the proposed approach may provide dynamic trust models for entities in the cyberspace. Moreover, the hybrid approach may be applied to low computing devices, such as IoT devices.

Peer-review: Externally peer-reviewed.

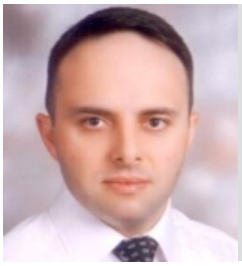
Conflict of Interest: The author have no conflicts of interest to declare.

Financial Disclosure: The author declared that the study has received no financial support.

References

1. W. Stallings, L. Brown, "Computer Security: Principles and Practice", 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2014.
2. P. Massa, "Trust in E-services: Technologies,Practices and Challenges", Idea Group Inc., 2007, ch. A Survey of Trust Use and Modeling in Real Online Systems, pp. 51-83. [CrossRef]
3. S. Bahtiyar, "Core-crust modeling approach for formal representation of trust in relation to computer security," Department of Computer Engineering, Bogazici University, İstanbul, Turkey, 2011.
4. T. Ryutov, "A socio-cognitive approach to modeling policies in open environments", in Eighth IEEE International Workshop on Policies for Distributed Systems and Networks, POLICY '07. Bologna, Italy: IEEE Computer Society, 13-15 Jun, 2007, pp. 29-38. [CrossRef]
5. O. Ajayi, R. Sinnott, A. Stell, "Trust realisation in multi-domain collaborative environments", in 6th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2007. Melbourne, Qld: IEEE, 11-13 Jul, 2007, pp. 906 - 11. [CrossRef]
6. S. Bahtiyar, M. U. Caglayan, "Extracting trust information˘ from security system of a service", Journal of Network and Computer Applications, vol. 35, no. 1, pp. 480-90, Jan, 2012. [CrossRef]
7. T. Grandison, M. Sloman, "A survey of trust in internet applications", IEEE Communications Survey, vol. 3, pp. 2-16, 2000. [CrossRef]
8. Z. Yan, "Trust management for mobile computing platforms", Department of Electrical and Communication Engineering, Helsinki University of Technology, Network Laboratory, 2007.
9. R. Juliana, P. U. Maheswari, "An energy efficient cluster head selection technique using network trust and swarm intelligence", Wireless Personal Communications, vol. 89, no. 2, pp. 351-64, Jun, 2016. [CrossRef]
10. D. Andert, R. Wakefield, J. Weise, "Trust modeling for security architecture development", Sun Microsystems, Inc., Santa Clara, CA, USA, Tech. Rep., 2002.
11. Z. Sun, Y. L. Han, K. J. R. Liu, "Defense of trust management vulnerabilities in distributed networks", IEEE Communications Magazine, vol. 46, pp. 112-9, 2008. [CrossRef]
12. M. Blaze, J. Feigenbaum, J. Lacy, "Decentralized trust management" in IEEE Symposium on Security and Privacy, ser. SP '96. Oakland, CA, USA: IEEE Computer Society, May 1996, pp. 164-73.
13. M. Blaze, S. Kannan, I. Lee, O. Sokolsky, J. M. Smith, A. D. Keremytis, W. Lee, "Dynamic trust management", IEEE Computer, vol. 42, pp. 44-52, 2009. [CrossRef]
14. N. Dimmock, A. Belokosztolszki, D. Eyers, J. Bacon, K. Moody, "Using trust and risk in role-based access control policies", in Proceedings of the ninth ACM symposium on Access control models and technologies, ser. SACMAT '04. New York, NY, USA: ACM, 2-4 June 2004, pp. 156-62. [CrossRef]
15. F. Olivieroa, L. Pelusoa, S. Romano, "Refacing: An autonomic approach to network security based on multidimensional trustworthiness", Computer Networks, vol. 52, pp. 2745-63, 2008. [CrossRef]
16. H. Zhu, S. Du, Z. Gao, M. Dong, Z. Cao, "A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks", IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 1, pp. 22-32, 2014. [CrossRef]
17. L. Xiao, Q. Yan, W. Lou, G. Chen, Y. T. Hou, "Proximitybased security techniques for mobile users in wireless networks", IEEE Transactions on Information Forensics and Security, vol. 8, no. 12, pp. 2089-100, 2013. [CrossRef]
18. L. M. Kaufman, "Can a trusted environment provide security?", IEEE Security&Privacy, vol. 8, no. 1, pp. 50-2, Jan/Feb 2010. [CrossRef]
19. G. Yang, C. H. Tan, "Certificateless cryptography with kgc trust level 3", Theoretical Computer Science, vol. 412, no. 39, pp. 5446- 57, 2011. [CrossRef]
20. X. Zhao, F. Zhang, "Fully cca2 secure identity-based broadcast encryption with black-box accountable authority", The Journal of Systems and Software, vol. 85, no. 3, pp. 708-16, Mar, 2012. [CrossRef]
21. U. Thiruvaazhi, R. Divya, "Web authentication protocol using zero knowledge proof", Information Security Journal: A Global Perspective, vol. 20, no. 2, pp. 112-21, Jan, 2011. [CrossRef]
22. F. Martinelli, M. Petrocchi, "A uniform framework for security and trust modeling and analysis with crypto-ccs", Electronic Notes in Theoretical Computer Science, vol. 186, pp. 85-99, Jul, 2007. [CrossRef]
23. S. Hajian, T. Tassa, F. Bonchi, "Individual privacy in social influence networks", Social Network Analysis and Mining, vol. 6, no. 1, pp. 1-14, 2015. [CrossRef]
24. W. Ni, M. Gu, X. Chen, "Location privacy-preserving k nearest neighbor query under user's preference", Knowledge-Based Systems, vol. 103, no. C, pp. 19-27, Jul, 2016. [CrossRef]
25. C. L. Miltgen, J. Henseler, C. Gelhard, A. Popovic, "Introducing˘ new products that affect consumer privacy: A mediation model", Journal of Business Research, vol. 69, no. 10, pp. 4659-66, Oct, 2016. [CrossRef]

26. D. Mazzei, G. Baldi, G. Fantoni, G. Montelisciani, A. Pitasi, L. Ricci, L. Rizzello, "A blockchain tokenizer for industrial iot trustless applications", Future Generation Computer Systems, vol. 105, pp. 423- 45, 2020. [CrossRef]
27. J. Li, J. Huai, C. Hu, "Peace-vo: A secure policy-enabled collaboration framework for virtual organizations", in 26th IEEE International Symposium on Reliable Distributed Systems, SRDS 2007, 2007.
28. K. R. B. Butler, S. Ryu, P. Traynor, P. D. McDaniel, "Leveraging identity-based cryptography for node id assignment in structured p2p systems", IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 12, pp. 1803-15, Dec, 2009. [CrossRef]
29. L. Lu, J. Han, Y. Liu, L. Hu, J. P. Huai, L. Ni, J. Ma, "Pseudo trust: Zero-knowledge authentication in anonymous p2ps", IEEE Transactions on Parallel and Distributed Systems, vol. 19, no. 10, pp. 1325-37, Oct 2008. [CrossRef]
30. S. Bahtiyar, M. U. Caglayan, "Trust assessment of security for e-health systems", Electronic Commerce Research and Applications, vol. 13, no. 3, pp. 164-77, May-Jun, 2014. [CrossRef]
31. P. B. Velloso, R. P. Laufer, D. de Oliveira Cunha, O. C. M. B. Duarte, G. Pujolle, "Trust management in mobile ad hoc networks using a scalable maturity-based model", IEEE Transactions on Network and Service Management, vol. 7, no. 3, pp. 172-85, 2010. [CrossRef]
32. R. Guha, R. Kumar, P. Raghavan, A. Tomkins, "Propagation of trust and distrust", in Proceedings of the 13th international conference on World Wide Web. New York, NY, USA: ACM, 17-22 May 2004, pp. 403-12. [CrossRef]
33. U. Kuter, J. Golbeck, "Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models", in Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence. Vancouver, British Columbia, Canada: AAAI Press, July 22-6, 2007, pp. 1377-82.
34. M. J. Probst, S. K. Kasera, "Statistical trust establishment in wireless sensor networks", in Proceedings of the 13th International Conference on Parallel and Distributed Systems, IEEE Computer Society, 2007, pp. 1-8. [CrossRef]
35. Y. L. Sun, W. Yu, Z. Han, K. J. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks", IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, pp. 305-17, 2006. [CrossRef]
36. E. Gray, J. M. Seigneur, Y. Chen, C. Jensen, "Trust propagation in small worlds", in Proceedings of the 1st international conference on Trust management. Springer-Verlag, 2003, pp. 239-54. [CrossRef]



Dr. Şerif Bahtiyar is an associate professor in the Department of Computer Engineering at Istanbul Technical University. He received his BS in Controland Computer Engineering and MS in Computer Engineering degrees bothfrom Istanbul Technical University in 2001 and 2004 respectively, and his PhD degree in Computer Engineering from Boğaziçi University in 2011. Dr.Bahtiyar was with MasterCard, TU- Berlin in Germany, and National ResearchInstitute of Electronics and Cryptology. His current research includes cybersecurity, mobile systems, trust modeling, and financial systems.