Heuritic Trust in IoT

Kaur, Bipjeet; Tange, Henrik

Published in:
Proceedings of the The Fifth International Conference on Wireless Communications, Vehicular Technology, Information Theory, Aerospace & Electronic Systems

Publication date:
2015

Document Version
Peer reviewed version

Heuristic Trust in IoT

Bipjeet Kaur
Center for Wireless Systems and Applications
Section for Information Technology
Technical University of Denmark
DK-2750 Ballerup
bika@dtu.dk

Henrik Tange
Center for Wireless Systems and Applications
Section for Information Technology
Technical University of Denmark
DK-2750 Ballerup
htan@dtu.dk

***Abstract*—Increasing use of Internet of things in diverse fields has demanding association in security and trust of such services. The trust on the services cannot exist independent. It is greatly depended on the security of system at each layer. In this paper we analyze different existing trust algorithms to develop a combined reputation and trust algorithm. This algorithm is developed in view of developing trustworthy services in the agriculture field where the end users are cautious about their sensitive data but keen in trustworthy services to improve the efficiency of the product.**

Keywords—Trustworthy services; reputation; ioGateway; OPC-UA

I. INTRODUCTION

Internet of things is widely distributed applications available for a variety of users. It has applications varying from domestic use over to wearable devices and the advanced use for example temperature controller of the boiler in thermal plant. The trust in most applications is required for users to guarantee on the reliability or availability of system services. For example, a distributed data storage application would want to guarantee that data stored by a user will always be available to the user with high probability and that it will persist in the network (even if temporarily offline) with a much higher probability [1]. Privacy along with reliability is needed so that the users whose data has been stored in a distributed system would guarantee to protect the content from being accessed by unauthorized users. There are several solutions to it with one easy solution is to encrypt all data before storing. However, in some applications access to unencrypted data is necessary for processing. It may also be sufficient to separating sensitive data from subject identities, or use legally binded strict privacy policies [2–5]. Anonymity as a specific application of privacy, users may only be willing to participate if a certain amount of anonymity is guaranteed. This may vary from no anonymity requirements, to hiding real-world identity behind a pseudonym, to requiring that an agent's actions be completely disconnected from both his real-world identity and his other actions. Obviously, a reputation system would be infeasible under the last requirement.

The paper is divided into three sections, namely Section: II Trust models in IoT, Section: III Trust model for Agricultural application and finally Section IV: Conclusion and future work.

II. TRUST MODELS IN IOT

Security is an important cornerstone for the Internet of Things (IoT). More specific, all common aspects of security must be regarded. With the huge amount of data created by IoTs, integrity of data and trust in the services offering the data is crucial. Further, to protect important data and user interests, confidentiality of data and privacy of users must be ensured. In addition to integrity and confidentiality, each request and response inside the IoT has to be authenticated in a proper and secure way. Trust is the subjective probability by which an individual expects that another individual performs a given action on which its welfare depends [6]. Reputation is the collected and processed information about one person/entity in a community from its former behavior as experienced by others.

According to Eschenauer et al. [12], trust is defined as "a set of relations among entities that participate in a protocol. These relations are based on the evidence generated by the previous interactions of entities within a protocol. In general, if the interactions have been faithful to the protocol, then trust will accumulate between these entities." Trust has also been defined as the degree of belief about the behavior of other entities (or agents) [13], often with an emphasis on context [14].

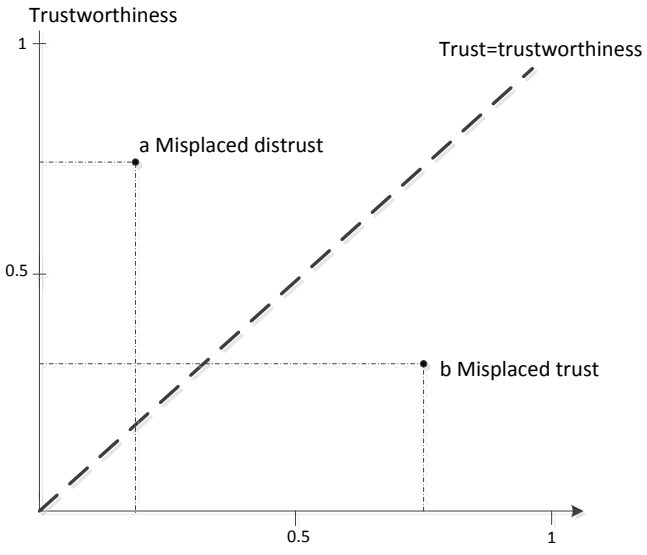

Figure:1Trust Level

Figure 1 explains how trust (i.e., subjective probability of trust level) and trustworthiness (i.e., objective probability of trust level) can differ. In Figure 1, the diagonal dashed line is assumed to be marks of well-founded trust in which the subjective probability of trust is equivalent to the objective probability [15]. In this Subjective and objective trust are supposed to be interchangeable entities. We have noticed that most of the current trust and reputation models in the literature follow these four general steps;

1. Collecting information about a certain participant in the community by asking other users their opinions or recommendations about that peer.

2. Aggregating all the received information properly and somehow computing a score for every peer in the network.

3. Selecting the most trustworthy or reputable entity in the community providing a certain service and effectively having an interaction with it, assessing a posteriori the satisfaction of the user with the received service.

4. According to the satisfaction obtained, a last step of punishing or rewarding is carried out, adjusting consequently the global trust (or reputation) deposited in the selected service provider [7].

A. Fuzzy Trust Model Description

An entity's trustworthiness is the quality indicator of the entity's services, which is used to predict the future behavior of the entity (stored in sensors or sensor-embedded things). Intuitively, if it is trustworthy enough, the entity will provide good services for future transactions. In most trust models, the domain of trustworthiness is assumed to be [0, 1]. Since the key issue in investigating fuzzy problems is to establish membership functions (membership degrees) by employing the fuzzy set theory, we have to create the mathematical model of fuzzy trust firstly [8].

Suppose that $SN = \{SN_1, SN_2, \ldots, SN_n\}$ is a problem domain of the fuzzy trust model. Here SN_i *(where* $i = 1,2,\ldots$ *,)* is a subset in the corresponding domain. Then we can get the following mapping,

$$MappingFunction: SN \times SN \rightarrow [0,1],$$
$$(SN_i, SN_j) \rightarrow \psi(SN_i, SN_j) \in [0,1]. \quad (1)$$

where $\psi(SN_i, SN_j)$ represents the degree of trust relationship between SN_i and SN_j. *MappingFunction* is a fuzzy relation mapping from $SN \times SN$ to [0,1].

In this scheme, a neighbor monitoring process is used to collect information of the package forwarding behaviors of the neighbors. Each sensor node in the network maintains a data forwarding transaction table *(DFT)* as follows:

$$DFT = < Source, Destination, RF_{i,j}, F_{i,j}, TTL > \quad (2)$$

where *Source* is the trust of evaluation evaluating nodes, *Destination* is the evaluated destination nodes, $RF_{i,j}$ denotes the times of successful transactions which node SN_i has made with node SN_j and $F_{i,j}$, denotes the positive transactions[8].

B. Reputation Evaluation:

Node SN_i evaluates the reputation of node SN_j with which it tries to make transactions by rating each package forwarding process as either positive or negative, depending on whether SN_j has completely done the transaction correctly.

Con denotes the evaluation of the whole metrics in order to judge whether this transaction is successful. The Con is computed by

$$Con = [EPFR, AEC, PDR].[\alpha\ \beta\ \gamma]^T$$
$$= \alpha.EPFR + \beta.AEC + \gamma.PDR \quad (3)$$

where $EPDR = \sum_i^k RECV_i / \sum_i^k SEND_i,\ 0 \le k \le n,$

$$AEC = = \sum_{i=1}^k consume_i / \sum_i^k SEND_i + RECV_i + \tau$$

and *PDR is packet delivery ratio[9]*.

where α, β, γ represent the corresponding aspect weights of the different resources. We also define a parameter $Sat_{Threshold}$ to describe the satisfaction degree. That means, if, then it indicates that node SN_i get a negative reputation evaluation to node SN_j ; if $Con \ge Sat_{Threshold}$, it indicates that node SN_i gets a positive reputation evaluation to node SN_j .

The reputation evaluation of all interactions from node SN_i to node SN_j is defined as follows:

$$\delta = F_{i,j} / RF_{i,j} \in [0,1] \quad (4)$$

Reputation evaluation is the basis of trust management. In our trust model, the reputation is evaluated considering three metrics, *EPFR, AEC* and *PDR.* Compared with other reputation evaluation methods, we consider more factors which can more accurately evaluate the behaviors of nodes according to specific characteristics of IoT. [9]

C. Eigen trust Algorithm:

A natural way to do this in a distributed environment is for peer *i* to ask its acquaintances about their opinions about other peers. It would make sense to weight their opinions by the trust peer *i* place in them:

$$t_{ik} = \sum_j C_{ij}\, C_{jk} \quad (5)$$

where t_{ik} represents the trust that peer *i* places in peer *k* based on asking his friends. It can be written in matrix notation: If C is defined as the normalized trust of matrix $[c_{ij}]$ and $\vec{t_i}$ to be vector containing the values t_{ik}, then $\vec{t_i} = C^T \vec{c_i}$. (Note that $\sum_j t_{ij} = 1$ which is the maximum value of trust of a network). The first thing that we can see when we want to compute the global trust vector $\vec{t}$

$$\boxed{\vec{t} = (C^T)^n \vec{c_i}} \quad (6)$$

is that this equation does not depend on *i*, because for all *i* the same vector is calculated. Such that c_i can be replaced with any distribution one can think, and because we want when starting all peers have the same chance, we can put the uniform distribution:

$$\boxed{\vec{t} = (C^T)^n \vec{e_i}} \quad (7)$$

where $e_i = 1/m$ for all *i*,

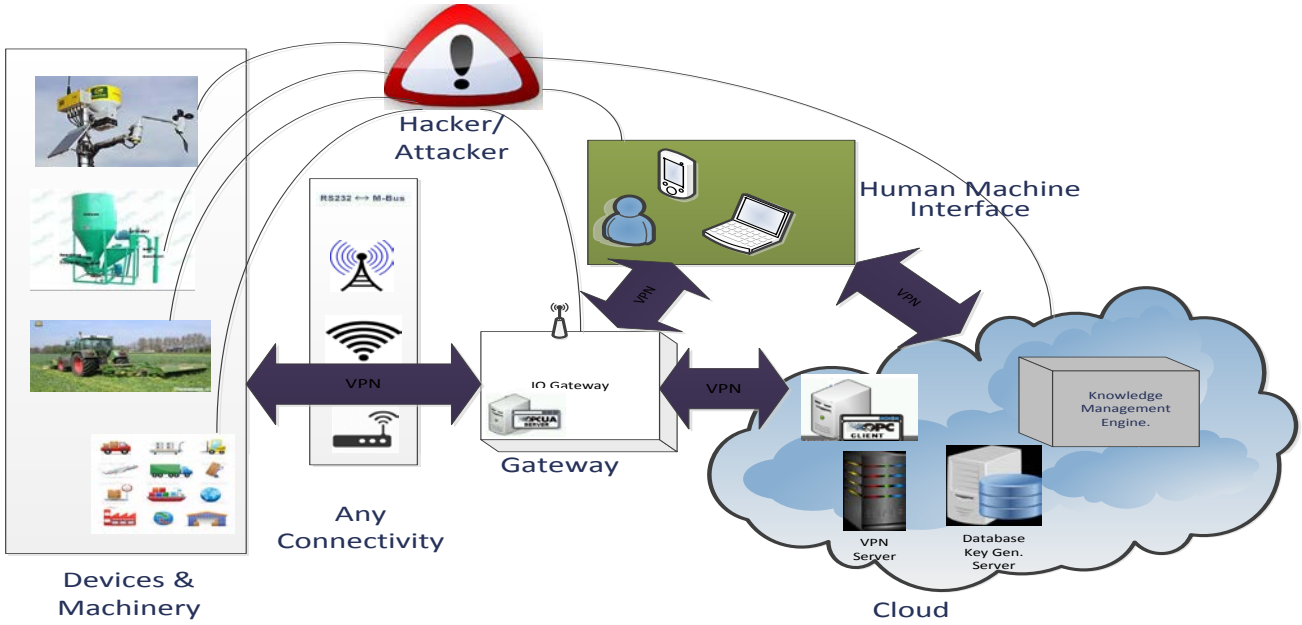

Figure2: Agricultural use case Architecture

and *m* is the number of peers in the network.
Secondly, it will not be a good idea to compute the n^{th} power of the matrix C, one reason being the fact that we don't know. By using the probabilistic interpretation of c_{ij} and looking at the PageRank computation it was found that the following algorithm works perfectly [10].

Simple EigenTrust Algorithm

$\vec{t^{(0)}} = \vec{e}$, $e_i = 1/m$

repeat

$\vec{t^{k+1}} = (C^T)^n \vec{t^k}$

$\delta = \| t^{k+1} - t^k \|$

until $\delta < \varepsilon$

There are some big drawbacks with this algorithm. The first is the lack of a prior notion of trust, then newcomers cannot gain trust (the Random-Surfer will get stuck when reaching a peer that has all $c_{ij} = 0$. Also a newcomer will not be known by the other peers). The third issue is that malicious peers can form groups or collectivities and in such situation the Random-Surfer can get stuck in these collectivities increasing the trust of these peers and decreasing the trust of all other peers. We address in more detail these problems:

• *Prior Notion of Trust*: In the algorithm above, it has been chosen to start with an uniform distribution over all peers, but this strategy doesn't hold if there can be malicious collectives, since if for example, one quarter of the network is forming such a collective, then there is a probability of ¼ to choose them in the beginning, getting stuck later. Also is not good to choose a peer *i* and start with $\vec{c_i}$. Thus we need to know a trusty peer to start with. A natural way to choose a peer or better a set of trusty peers is to consider the peers who first joined the network. We will call these peers pre-trusted peers, though these peers will not have special duties. The only thing that is required is that there are peers in the network that can be trusted, and they will never change their side, becoming malicious peers. The peers who join first (or build) the network have no interest to subvert the network.

Thus if there are P pre-trusted peers, distribution $\vec{p}$ can be on them by setting:

$$\begin{cases} p_i = 1/p & \text{if } i \in P; \\ = 0 & \text{otherwise} \end{cases} \qquad (8)$$

III. TRUST MODEL FOR AGRICULTURAL APPLICATION

The trust model is developed for agricultural services. The basic architecture of this application includes the sensors from weather station, such as temperature, moisture, rainfall, wind speed, etc., machinery like cattle feeding machine, tractor, harvester, etc., which may be connected by OPC-UA protocol or RS232 by other sensor systems, even the logistic system which include the transportation medias which may be connected by cellular, 3G, 4G, GSM network, giving information regarding location, temperature of food products, etc. All this data is sent by various connectivity media such as WiFi, GSM, 3G, 4G, Bluetooth, ZigBee, etc. to a gateway via VPN. This gateway is further connected the cloud which has Database server, Knowledge Management Engine which also receive inputs from various external sources such as FMIS (Farm Management Information Systems), advisory services such as vet nary services, farm disease control services, etc.

and VPN server for the security services. The cloud also contains OPC-UA client, to connect the gateway OPC-UA server. The cloud is connected to the handheld devices such as tablets, PDA, laptops via VPN.

The whole architecture is associated with trustworthy services. But the trust as discussed above is fully dependent on the security of this architecture. Fig.2 depicts the places of attacks by malicious user or hacker in the chain. The security is maintained using various security algorithms at different layers in the chain as discussed below:

1. At the sensor or the physical layer the sensor is secured by enabling the standard security protocols provided by the communication protocol of the sensor for example Bluetooth 4.0, ZigBee, 3G etc.
2. The communication between the Gateway to the sensory systems and machinery and the cloud, along with cloud and Human Machine Interface devices is done through the VPN connection.
3. The Machinery systems are connected to the Cloud through Gateway via OPC-UA which has its own security protocol which is enabled to make this connection secure.
4. During setup of a new node, a NFC chip close to the sensor is used to setup the sensor using for instance communicating via a smart phone to central system in cloud.
5. Communication must always secured between the sensor and the central system (the GW is only carrying data)
6. Furthermore, the system must add an algorithm which prevents logging of not-trustworthy data (for instance physical manipulated temperatures).

In the development of trustworthy services for the agricultural domain, the algorithm designed is such that it used the combination of Reputation and the EigenValue algorithm.

PRIME project [17] suggest: Overall trust value (0-1) is a function of the reputation R and the individual trust of the information or the data source:

$$T=\int\phi(R,t) \qquad (9)$$

where t is the time elapsed since the reputation R was last modified[16]. Each service provided by the cloud is given service trust score integrated with data and this trust is given as an output to the user as low trust value, medium trust value and high trust value (for example some disease prediction model and N Fertilization). We identify the following data score storage options:

1. Each sensor data is associated with trust value ranging between [0,1]. The lowest trust value is zero and highest 1.
2. The gateway collects data from the sensors and is validated before being stored in the Gateway OPC UA address space. The values which cannot be validated or have very low range trust values are discarded at this place.
3. Cloud databases where both original readings and aggregated data may be stored.
4. Distributed storage, when separate trust scores are saved both in primary acquisition place and Cloud databases.

Field data like temperature, soil moisture, rainfall, GPS coordinates, etc. can be acquired by an intelligent sensor or measured by Gateway itself. Though intelligent sensor system like weather stations can calculate trust score, but most probably the first place in the whole sensor data life cycle suitable for electronic storing is Gateway. It handles heterogeneous data in nodes of OPC UA address space. Time stamps are also associated with each sensor reading in OPC UA address space.

The data validation algorithm in the gateway includes the historical data, the time stamp and the other information for example the validation of the temperature in the field can be validated using the GPS co-ordinate along with predicted values from general knowledge and also from the meteorological information. If the value is validated as true it is given trust value as given value:

Trust value of each source: $TV_1, TV_2, TV_3, \ldots TV_N$

where N is the no of sensors in the data fusion model.

A trust score determined inside cloud or the gateway will most probably be valid until its reset or power off (stored in volatile memory). In this case data trust score of each node representing sensor readings can be refreshed at power up of gateway by executing self-check procedures. It can be done by protocol adapter or processing adapter. Gateway processing adapter is adding software components responsible for local data preprocessing or aggregation.

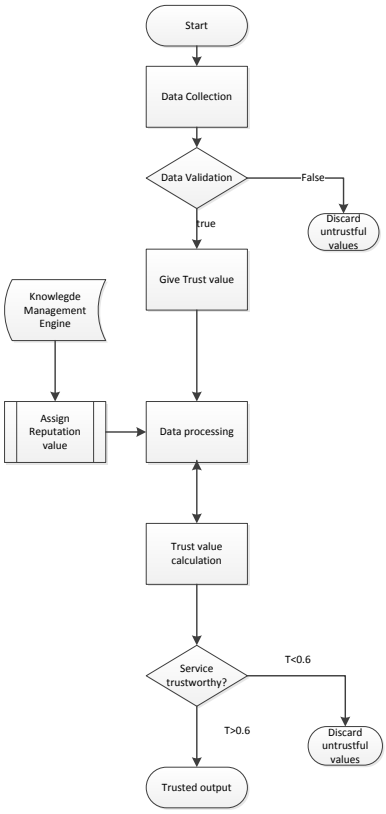

Figure: 3 Flow Chart of Combined Algorithm

The Knowledge Management Engine gets information from different sources like FMIS, Legal and regulatory documents which are allocated reputation value as shown as follows:

External information sources: ES_1,ES_2, ES_3,…..ES_M

where *M* is the no of external information source data taken in the data fusion model. These values are then taken by the data fusion model and overall function calculates the total trust value of the service. The description of the function is beyond the scope of this paper. Derived from statistical or other type of processing, for example, determination of outliers, comparing the output with other available similar services is used for improving the algorithm function. Trust services are activated to analyze of acquired data patterns and derivation of trust scores. Discarding values with lower overall trust will increase the overall trust in the system chain and the services provided by the system. Technically speaking, it is easier to save derived trust value in the Cloud database than to download derived trust value to OPC UA address space of Gateway.

IV. Conclusion and Future Work

Model has been tested for agricultural services such as the early disease alarm model in various fields for real time data and for the N fertilization with the simulated data. The trust algorithm has yet to be optimized better performance in case of the efficiency and delay. It needs to improve in future with statistical analysis and along with comparing data from other locations considering different geographical locations farms without exposing the source. It can also improvise by using principles from Big Data comparison techniques and extending the model to other applications such as logistic services etc.

Acknowledgment

This project is carried out in close collaboration with Prof. Dr. Joseph Kueng, Institute for Application Oriented Knowledge Processing, JKU, Linz. This research was funded by the FP7 - EU FRAMEWORK PROGRAMME under grant agreement No. 604659 (project CLAFIS).

References

[1] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, B. Zhao, OceanStore: An Architecture for Global-scale Persistent Storage, in: Proceedings of ACM ASPLOS, ACM, 2000.

URL citeseer.nj.nec.com/kubiatowicz00oceanstore.html

[2] [11] P. Maniatis, M. Roussopoulos, T. Giuli, D. S. H. Rosenthal, M. Baker, Y. Muliadi, Preserving peer replicas by rate-limited sampled voting, in: 19th ACM Symposium on Operating Systems Principles (SOSP 2003), 2003.

[3] M. K. Reiter, A. D. Rubin, Crowds: Anonymity for web transactions, in: ACM Transactions on Information and System Security, 1998.

[4] G. Agarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, N. Mishra, R. Motwani, U. Srivastava, D. Thomas, J. Widom, Y. Xu, Vision Paper: Enabling Privacy for the Paranoids, in: VLDB, 2004.

[5] G. Agarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, Y. Xu, Two Can Keep a Secret: A Distributed Architecture for Secure Database Services, in: CIDR, 2005.

[6] Gambetta, T.: Can we trust trust? In: D. Gambetta (Ed.), Trust: making and Breaking Cooperative Relations, Basil Blackwell, Oxford, 213-238. (1990)

[7] Sergio Marti, Hector Garcia-Molina, Taxonomy of Trust: Categorizing P2P Reputation Systems, Stanford University {smarti, hector}@cs.stanford.ed

[8] Azzedin, F., Ridha, A., Rizvi, A.: Fuzzy trust for peer-to-peer based systems. In Proc. of World Academy of Science, Engineering and Technology, Vol. 21, 123- 127. (2007)

[9] Dong Chen1 , Guiran Chang2 , Dawei Sun1 , Jiajia Li1 , Jie Jia1 , and Xingwei Wang, TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things, ComSIS Vol. 8, No. 4, Special Issue, October 2011

[10] David Challener, Kent Yoder, Ryan Catherman, David Safford, Leendert Van Doorn, A Practical Guide to Trusted Computing

[11] Adrian Alexa, Reputation Management in P2P Networks: The EigenTrust Algorithm.

[12] L. Eschenauer, V. D. Gligor, and J. Baras, “On Trust Establishment in Mobile Ad Hoc Networks,” Proc. 10th Int’l Security Protocols Workshop, Cambridge, U.K., Apr. 2002, vol. 2845, pp. 47-66.

[13] L. Capra, “Toward a Human Trust Model for Mobile Ad-hoc Networks,” Proc. 2nd UK-UbiNet Workshop, 5-7 May 2004, Cambridge University, Cambridge, UK.

[14] H. Li and M. Singhal, “Trust Management in Distributed Systems,” Computers, vol. 40, no.2, Feb. 2007, pp. 45-53.

[15] Solhaug, D. Elgesem, and K. Stolen, “Why Trust is not proportional to Risk?” Proc. 2nd Int’l Conf. on Availability, Reliability, and Security (ARES’07), 10-13 April 2007, Vienna, Austria, pp. 11-18.

[16] Christer Andersson , Jan Camenisch, Stephen Crane , Simone Fischer-Hübner, Ronald Leenes, Siani Pearson , John Sören Pettersson, and Dieter Sommer, Karlstad University, Sweden; Trust in PRIME, 2005 IEEE International Symposium on Signal Processing and Information Technology

[17] John Sören Petterson, Simone Fischer-Hübner, Jenny Nilsson, Ninni Danielsson, Mike Bergmann, Thomas Kriegelstein, Sebastian Clauß, Henry Krasemann: Making PRIME Usable. In Proceedings of the Symposium of Usable Privacy and Security (SOUPS), 4-6 June 2005, Carnegie Mellon University, Pittsburgh, PA, USA. July 2005.