TRUST IN M2M COMMUNICATION

Addressing New Security Threats

Inhyok Cha, Yogendra Shah, Andreas U. Schmidt,
Andreas Leicher, and Michael Victor (Mike) Meyerstein

Machine-to-machine (M2M) communication is viewed as one of the next frontiers in wireless communications. Freed from the traditional constraint of wireless devices that require manning or human intervention, communication using M2M equipment (M2ME) is expected to open up exciting new use cases, services, and applications, with benefits for the general masses and market opportunities for various stakeholders such as manufacturers of M2ME and components, service providers, and communication network operators.

Considering the large number of M2MEs expected to be deployed in highly distributed networks and because of the requirements for low-cost devices and implementations, global enforcement of security will not be practical. As the conventional centralized IT network security model, protected by a firewall, becomes challenged by the need for a dispersed model, decentralized methods for establishing security are being explored. The growing

Digital Object Identifier 10.1109/MVT.2009.933478

AN ENTITY CAN BE TRUSTED IF IT PREDICTABLY AND OBSERVABLY BEHAVES IN A MANNER EXPECTED FOR ITS INTENDED PURPOSE.

trend toward decentralized systems produces numerous situations in which enforcement, by practical necessity, has to be complemented by controlled risk. Principles of enforcement embraced by traditional concepts of access control policies are being supplemented by trust. An entity can be trusted if it predictably and observably behaves in a manner expected for its intended purpose. By delegating parts of the enforcement tasks to trusted elements dispersed in a system, transitive (i.e., multihop) trust relationships can be established. This evolved security model, balancing trust and enforcement, results in a useful, practical, and scalable approach for M2M communication security, which is a critical factor for the overall success of the M2M market.

M2M Use Cases

Various standards organizations have identified a number of use cases for M2M communication [1], [2]. This section describes some of the use cases, covering important user requirements to clarify the security requirements on M2M systems. All these cases have some common security requirements. Since devices are typically unmanned and a high value is placed on the information handled and communicated by these devices, information security and trustworthy operation of these devices needs special emphasis, as does the ability to manage the M2ME over the air.

1) *Traffic cameras* with cellular connectivity may be installed in locations such as motorway overpasses or remote stretches of roadway. Cameras may also require simultaneous secure local wireless local area network (WLAN) connectivity to the next camera down the road, e.g., when measuring average speed.
2) *Car rental* agencies may provide their customers with vehicles equipped with on-dash mounted voice and multimedia and Internet-access mobile communication system. Such a M2ME will be mounted, e.g., by the dealer with no prior knowledge of the customer's cellular network subscriptions. Desired features for such a M2ME may include remote, over-the-air methods to change, on demand and per human customer, the subscription credential to be used by the M2ME to access cellular networks. This use case requires assurance by the user and his home network operator that subscriptions are active in the car only while the rental contract is valid but then are deactivated when the rental contract expires.
3) *Asset/cargo tracking* systems allow owners or users of equipment to monitor critical parameters, perform remote commands, or monitor movements. Asset and cargo tracking will often require that the M2ME be placed in areas where physical access is difficult. Such placements would be part of a service provider's attempt to protect it from the environment and resist theft and tampering of the M2ME. This placement, together with the fact that the M2ME is likely to be highly mobile, can make it difficult and costly for the owner to physically access the M2ME.

Security Threats for M2M

M2MEs have unique characteristics and subscription and deployment contexts [1]. M2MEs are typically required to be small, inexpensive, able to operate unattended by humans for extended periods of time, and to communicate over the wireless area network (WAN) or WLAN. M2MEs are typically deployed in the field for many years, and after deployment, tend to require remote management of their functionality. They also require flexibility in terms of subscription management. In addition, in many use cases, it is likely that M2MEs will be deployed in very large quantities, and many of them will also be mobile, making it unrealistic or impossible for operators or subscribers to send personnel to manage or service them. These requirements introduce a number of unique security vulnerabilities for the M2MEs and the wireless communication networks over which they communicate. In [2], the Third Generation Partnership Project (3GPP) Security Workgroup (SA3) has collected categories of vulnerabilities:

1) *physical attacks* including the insertion of valid authentication tokens into a manipulated device, inserting and/or booting with fraudulent or modified software (reflashing), and environmental/side-channel attacks, both before and after in-field deployment
2) *compromise of credentials* comprising brute force attacks on tokens and (weak) authentication algorithms, physical intrusion, or side-channel attacks, as well as malicious cloning of authentication tokens residing on the machine communication identity module (MCIM)
3) *configuration attacks* such as fraudulent software update/configuration changes; misconfiguration by the owner, subscriber, or user; and misconfiguration or compromise of the access control lists.
4) *protocol attacks* directed against the device, which include man-in-the-middle attacks upon first network access, denial-of-service (DoS) attacks, compromising a device by exploiting weaknesses of active network services, and attacks on over-the-air management (OAM) and its traffic
5) *attacks on the core network*, the main threats to the mobile network operator (MNO), include impersonation of devices; traffic tunneling between impersonated devices; misconfiguration of the firewall in the modem, router, or gateways; DoS attacks against the

core network; also changing the device's authorized physical location in an unauthorized fashion or attacks on the network, using a rogue device

6) *user data and identity privacy attacks* include eavesdropping user's or device's data sent over the access network; masquerading as another user/subscriber's device; revealing user's network ID or other confidential data to unauthorized parties.

Some of the vulnerabilities that are more specifically geared to the subscription aspects of the M2ME are exhaustive and span the network, device, and user [2]. However, for special application contexts, such as vehicular communication, more specific requirements need additional consideration. For example, in the case of vehicular ad hoc networks (VANETs), there are issues of liability identification, i.e., restricting user privacy to allow for identification of users whose actions disrupt the operation of nodes or the transportation system. Also, in-transit data, traffic tampering (e.g., to pass a toll point without paying) and onboard tampering (e.g., with sensors for velocity or location of the vehicle) may be easier than tampering with the M2ME itself [3].

The Trusted Environment

To establish trust relationships in dispersed systems, the systems must contain security-relevant elements and capabilities to form a trust boundary. These components include methods to extend the trust boundary and convey trust to an external entity. A trusted environment (TRE) provides a hardware security anchor and root of trust, allowing for the construction of systems that combine the characteristics of trust and enforcement.

The TRE is a logically separate entity within the M2ME, containing all necessary resources to provide a trustworthy environment for the execution of software and storage of sensitive data. The TRE provides isolation of software and stores data by separating them from the rest of the M2ME, thus protecting from unauthorized access. The TRE provides a trust anchor, which is secured against tampering by hardware security measures. In particular, it provides the root of trust (RoT) for secure operation. The RoT is an immutable part of the TRE, which secures internal operation and is able to expose properties, or the system's identity, to external entities. Based on the RoT, the TRE performs a secure start-up process ensuring that the TRE reaches a determined trustworthy state. The secure start-up includes all components and programs that are executed during the system boot and can be extended to the operating system and software, thus expanding the trust. A model for this extension process is the verification of every new component when it is loaded, by measuring its integrity [4]. This method uniquely identifies every component, its state and configuration. The measurement can then be compared to reference values, and the verification entity can then decide whether to include this new component in the extended trust boundary or not. As verification is intended to take place locally, it relies on the TRE being in a predefined state after a completed verification process. Validation, denoting the ability to technically assess the state of a system for all security-relevant properties, requires that a reporting entity transfers the results of verification to an external party. The external validator can then assess the device's state. Validation makes the M2ME's functions observable and, thus, trustworthy. In addition, the TRE can provide protected functions for the authentication of the M2ME toward the network, e.g., by storing the authentication data inside the TRE.

M2M COMMUNICATION APPLICATIONS AND SCENARIOS ARE GROWING AND LEAD THE WAY TO NEW BUSINESS CASES.

The TRE provides cryptographic capabilities including symmetric and asymmetric encryption and decryption, hash value calculation and verification, random number generation (RNG), and digital signature capabilities. In addition, secure storage for keys, credentials, and authentication data must be provided by the TRE. The storage area may be outside the TRE but protected by it, e.g., by encrypting with a key stored inside the TRE. The TRE must also be able to establish secure communication channels with other parts of the M2ME. Interfaces for this are initialized in the secure start-up process, integrity-protected by the TRE and, hence, can be assumed to operate correctly. Two interface categories can be distinguished. Protected interfaces provide integrity protection and/or confidentiality of the data carried across them. This can be achieved by the use of security protocols or hardware interfaces. Further functionalities such as entity and message authentication can be provided by security protocols. Unprotected interfaces facilitate communication between the TRE and general resources of the M2ME. These unprotected interfaces can, nevertheless, give access to data that are cryptographically protected by the TRE. Even unprotected interfaces can benefit from other security measures such as authorization or making the interface available only after the TRE checks the code of its counterpart across the interface. Figure 1 shows the components and interfaces of the TRE in a M2ME.

Verification of Trustworthiness of M2ME

Practical requirements and threats of M2M application scenarios have two main aspects: 1) unpredictable connectivity to the core network and 2) demand for high configurability and flexibility of the M2ME. Both aspects arise at the time of specification of an M2ME and its interaction with the network and must be taken into account early on to enable a broad range of use cases with optimal cost-efficiency. We view fulfilling 2) under the condition 1) as

the main obstacle for the takeoff of the M2M market. Security is essential for this problem and amounts to satisfying two concrete protection goals:

- ensure that M2ME can reach and operate in, locally, a secure state without network connectivity
- enable the establishment of assurance, locally and remotely, concerning the state of the M2ME, to assess its security properties and, hence, trustworthy operation.

Perhaps, the most important role is played by these protection goals when the states of an M2ME are changed in a controlled way by OAM or local management. Network operators and M2ME owners have a clear mutual interest to have independent abilities to validate an M2ME's state in such a case. Arguments for dedicated means to validate M2MEs are applicable to the whole life cycle, from initial deployment (for installation and verification), maintenance (for diagnosis and success verification), to OAM (validation of correct configuration changes).

The means to validate a system in its operational phase are different from predeployment testing and certification, or formal security proofs on a design, leading into the realm of trusted systems and trusted computing. We can distinguish three main variants. We name autonomous validation (AuV) as a model of closed systems that arrive at a secure state merely by local means and do not communicate state information to the exterior. At the other extreme, remote validation (RV) is an abstraction of what the trusted computing group (TCG) has specified as remote attestation, where the M2ME reports state information in a secure way. A broad spectrum of other variants lies in between the range marked by these extremes.

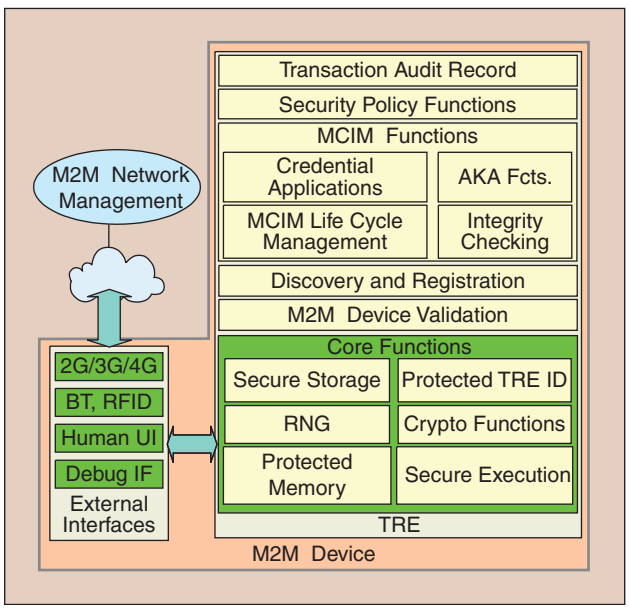

FIGURE 1 Components and interfaces of TRE in a M2ME. AKA: authentication and Key agreement; ID: identity; BT: Bluetooth; RFID: radio frequency identification; UI: user interface; IF: interface.

We call this spectrum semiautonomous validation (SAV). There exists one concrete example on the level of technical specifications: secure boot, as specified by the TCG's Mobile Phone Working Group [4].

It should be noted that, for the understanding of the following, AuV and RV are already technically realized (via smart cards and TCG remote attestation), whereas SAV is a spectrum of technology options that remains to be explored. In this view, SAV is a competing option, which we believe to have many advantages over AuV and RV. We describe the salient (anticipated) pros and cons of the three options in the next subsections, where we give more details on their technical characteristics.

Autonomous Validation

AuV is a procedure that does not depend upon external entities, and local verification is assumed to have occurred before the M2ME attempts communication with the exterior or perform other critical operations. Local verification is assumed to be absolutely secure, as no direct evidence of it is provided to the outside world. AuV lays all enforcement duties on the implied trust in the device and applies a closed, immutable system model, essentially that of smart cards. Validation by a relying party is then implicit, e.g., during network attachment. A typical example is the release of an authentication secret by a smart card.

Security resting only on devices has been broken in the past and is more likely to be broken as devices become open-computing platforms. AuV delivers no information for advanced security requirements: in particular, if parts of the device outside of the immutable TRE component's boundary are compromised, the TRE might not be aware of that and the external world cannot gain knowledge about its state other than by inference from its behavior toward the network. Labeling of rogue devices can therefore be impossible. AuV may be realized in such a way that verification is reactive to certain conditions, e.g., by closing the device down and going to reboot. Use cases that essentially require connectivity, such as an M2ME for theft protection, are disabled if an attacker prevents network connection by tampering with M2ME configurations. Compromised devices could only be recovered by costly in-field replacement. Remote management is also difficult. Specifically, there may be a loss of security in software download and installation since it potentially delivers software and secrets to rogue devices. Thus, AuV is prone to entailing out-of-band maintenance. A lot of burdens and risk rest with the owners of such M2MEs, but the network also bears an additional burden because it has to keep track of the state of every AuV device. If multiple parties can force updates on a device, this may become complicated. Finally, with AuV, the freshness of the information on local verification is not guaranteed. As AuV is likely to happen infrequently in practice, the M2ME's state may change significantly during its operation, in an unobservable manner, allowing an attacker

to introduce malicious software. AuV is prone to this kind of timing attack.

Remote Validation

In RV, the relying party directly assesses the validity of the device based on the evidence for the verification received. Local verification is only passive, just measuring integrity values of the loaded and started components [5]. A stored measurement log (SML) must be conveyed to the relying party, who makes all policy decisions. In a remote attestation, a TCG-trusted platform exhibits a SML and platform configuration register (PCR), in a signed message to the relying party. The signing keys are ephemeral, certified by a privacy certification authority (PCA), which acts as an identity provider for the purpose of validation. Pseudonymity of remote attestation may not be sufficient in all cases. TCG has additionally defined direct anonymous attestation (DAA) [6], [7] based on zero-knowledge proofs [8].

RV, as represented by remote attestation, poses practical problems with respect to scalability and complexity, as it lays the full computational load on (central) access points to networks. The RV of an SML can be costly for open platforms with many software components in numerous versions and configurations. RV requires an enormous database of reference values, namely, reference integrity measurements (RIMs), and an infrastructure to let stakeholders define the desired target configurations of devices. This makes remote management of a device impractical with RV. Finally, RV of complex open devices compromises privacy, despite usage of a PCA, as the revealed SML might be almost unique to a device. Some of the disadvantages might be alleviated by refined forms of remote attestation such as semantic [9] or property-based attestation [10], [11], aimed at exhibiting the characteristics of components, rather than a concrete implementation. These options, however, need more research before they may become practical.

Semiautonomous Validation

SAV is any procedure whereby the device validity is assessed within itself during verification and policy decisions are made and enforced during this local verification. However, in this case, the result and required evidence are signaled to the relying party who can make its own decisions based on the content of the validation messages and, optionally, additional information from trusted third parties who provide security properties of the M2ME. A model case for SAV is secure boot, followed by signaling of the so-called event structure [4], security properties of the device and, optionally, indication of measurement values to the relying party. SAV symmetrically distributes verification and enforcement tasks between the device and relying party. Specifically, in secure boot, the former makes decisions at load time of components, whereas the latter can enforce decisions on interactions permitted to the M2ME based on the state evidence provided.

M2MEs are typically deployed in the field for many years, and after deployment, tend to require remote management of their functionality.

SAV may be a promising avenue to remedy the disadvantages of the AuV and RV. It can potentially transport the validation information more efficiently than RV, in the form of indicators of the measurement values used in verification. This can also be used to protect privacy, e.g., when such an indication designates a group of components with the same functionality and trustworthiness (such as versions). The interplay of local and remote enforcement in verification and during validation also opens options for OAM. On the path to technical realization of such opportunities, the Trusted Network Connect (TNC) Working Group of the TCG has introduced the concept of remediation ([12], p. 24) to obtain "support for the isolation and remediation of access requestors (ARs), which do not succeed in obtaining network access permission due to failures in integrity verification." This allows "to bring the AR up to date in all integrity-related information, as defined by the current policy for authorization. Examples include O/S patches, antivirus (AV) updates, firmware upgrades, etc." ([12], p. 25). Concrete concepts for realization of OAM will have to rely on an infrastructure for the efficient representation and communication of RIM information. TCG MP Working Group has started to define such services for mobile devices [4], in particular, to ingest RIMs for verification. TCG Infrastructure Working Group is establishing a generic architecture and data structures for validation [13]. However, more research and development is needed to devise efficient and effective SAV on this path. RIM certificates play an important role in SAV. They are provided by a certification authority that has assessed the corresponding component. Certification methods and bodies can be diverse and lead to different levels of operational trustworthiness, entailing further flexibility for a relying party who gets more fine-grained information on the device. SAV is also the only practical option for systems that are resource limited so that they lack either the security of a closed system needed for AuV or memory and communication capabilities to perform the extensive reporting needed for RV.

From the viewpoint of technically verifying the different solutions AuV, RV, and SAV, and evaluating their viability in the field, we described the most important features that are touchstone for real-world implementations. For security issues, AuV relies completely on manufacturer certification of M2ME, while both RV and SAV give fine-grained control over trust in specific components and component manufacturers. The practical efficacy of both RV and SAV for M2ME validation and management remains to be proven.

ONE IMPORTANT PILLAR OF SUCH A SHIFT WILL BE A NEW, MORE BALANCED MIX OF DEVICE-CENTRIC TRUST AND TRADITIONAL ENFORCEMENT OF SECURITY PROPERTIES.

Validation and Enforcement

Validation and local verification are the central conceptual link between trusting a device and enforcing policies on its behavior. Based on the results of validation, policy decisions can be made, and verification, in turn, can incorporate enforcement on secure start-up. Figure 2 shows a simplified picture for policy enforcement through validation. More details can be found in [14]. A common trait of all variants is that the device needs a policy information point (PIP) to support the validity decision by the relying party. The PIP in the device performs the measurement and securely records the results. Since validation is always performed for a purpose, there is a policy enforcement point (PEP) present at the relying party. Based on the validation information, it can enforce decisions such as granting network access. The richness of the information varies significantly between the variants.

In RV, the device must transmit the full SML to the relying party and information binding it to the device state and protecting authenticity. The relying party's PIP must contain a database of possible allowed device states including RIMs. Based on this, the policy decision point (PDP) at the validator retraces the SML, e.g., recalculates digest values. The PEP obtains a graded result from this, stating up to which position in the SML the M2ME was trustworthy. On this information, the PEP acts, e.g., by (dis)allowing network access.

In AuV, all functionality for measurement, verification, and enforcement during secure boot and runtime is localized in the device's PIP, PDP, and PEP, respectively. The relying party's PEP can enforce only policies based on the static information contained in the signaling at first network contact, e.g., system type or identity. Since validation information is not present, no validation-specific PIP and PDP are used at the validator. However, a PIP and PDP can be constructed, in this case, based on TS identities and connection history—in effect, a traditional authentication, authorization, and accounting (AAA) system [15].

SAV allows for policy systems on both sides. The key to this is a codification of validation data, which may consist in a concise event log containing, essentially, references to RIMs and associated certificates (the precise content may depend on implementation requirements). This abstraction is made possible by the device's PDP, which, at the time of verification, makes the association of component to target RIM. For that, it relies on an internal, protected, RIM database, whose management adds to the functional role of the PIP (beyond measurement). Attestation to codified RIMs allows interaction with the relying party in validation. The PDP of the relying party can use its own RIM database (provisioned by its PDP) to compare the attested state with fine granularity to a desired state. The PEP can thus initiate 1) provisioning of new RIMs to the device, 2) unload of undesired components, 3) load of new, desired components, and finally, 4) updates of components. These processes are captured by the term *remediation*. To show the success of the remediation, the device needs to revalidate only using the new part of the event log. From the viewpoint of policy systems, RIMs add an essential piece to enable general policies for validation: A codified ontology on which conditions can be evaluated and decisions taken.

Conclusions

M2M communication applications and scenarios are growing and lead the way to new business cases. Because of the nature of M2M scenarios, involving unguarded, distributed devices, new security threats emerge. The use case scenarios for M2M communication also address the new requirement on flexibility, because of deployment scenarios of the M2ME in the field. We believe that these new requirements require a paradigm shift. One important pillar of such a shift will be a new, more balanced mix of device-centric trust and traditional enforcement of security properties. Distributing trust building and enforcement tasks between device and network leads to scalable concepts, which can be adapted flexibly to the technical tasks. The two most important

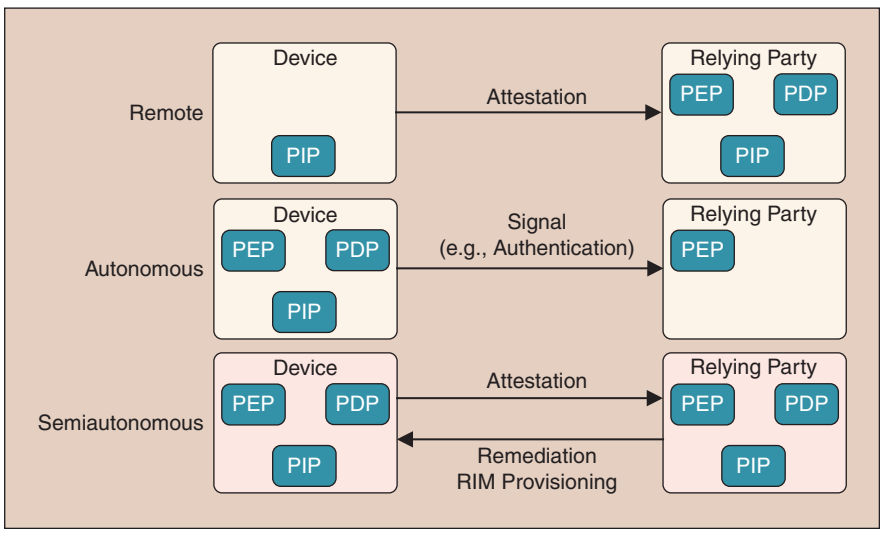

FIGURE 2 Mapping variants of validation to policy enforcement.

building blocks for this are local state control, via secure boot, and conveying trust by SAV. By embracing these advanced concepts of security, the unique needs of the M2M market, i.e., remote management of M2ME and subscription management can be met [16]. The presented ideas enable new business opportunities aligned with the goal of achieving hardware-backed security.

Author Information

Inhyok Cha (inhyok.cha@interdigital.com) has graduated from Seoul National University, with a B.S. degree in 1988 and an M.S. degree in 1990, and a Ph.D. degree in electrical engineering from the University of Pennsylvania in 1995. He did R&D and cellular wireless R&D management for Lucent Technologies between 1996 and 2003. Since 2004, he has been with InterDigital Communications Corporation. His interest includes M2M communications, wireless communication security, and trusted computing. He is an author of a number of journal and conference papers and an inventor with many patent awards.

Yogendra Shah (yogendra.shah@interdigital.com) earned his B.Sc. and Ph.D. degrees in electrical engineering from The City University, London, in 1982 and 1985, respectively. He has worked in the wireless industry developing consumer products, incorporating wireless technologies, and has worked as a systems engineer and product developer at various organizations before joining InterDigital. He is currently a principal engineer in the CTO Office, with research interests in developing advanced communications modem technologies and wireless security technologies. He is an inventor or coinventor of several U.S. patents and has been awarded the President's and CTO Innovation awards at InterDigital.

Andreas U. Schmidt (andreas.schmidt@novalyst.de) received his doctorate in mathematics at the University of Frankfurt/Main in 1999. After research stays in Durban and Pisa, he became a senior researcher at the Fraunhofer Institute for Secure Information Technology, Darmstadt. He currently heads the security area of CREATE-NET, Italy. His research interests include applications of trusted computing in the mobile domain, identity management, privacy, voice over IP security, information economy, long-term security, and secure transactions. He produced more than 50 publications in various fields, works as reviewer for renowned journals in security, and served in the program committee of numerous conferences. He organizes the conference MobiSec on mobile security.

Andreas Leicher (andreas.leicher@novalyst.de) received his diploma in computer science at Frankfurt University in 2009. Having his main focus on IT security, he developed a framework for trusted computing applications, implementing a trusted ticket system by enhancing the Kerberos authentication protocol. He is working as a consultant with Novalyst IT. His interests are in the areas of trusted computing, IT security, privacy, identity management, mobile systems, and 3GPP.

Michael Victor (Mike) Meyerstein (meyersmv@btinternet.com) graduated from the University of Liverpool in 1971 with an honors degree in mechanical engineering science. He worked in production engineering and then in the U.K. nuclear energy industry. From 1978 to 1985, he worked in Canada on the development and pilot production of mainframe disk drives and mechanical plant for telecoms. From 1985 to 2002, he worked in the U.K. telecoms industry, mostly on smart cards and information security. Since 2002, he has been an independent consultant in that field, and his client list includes British Technology Group, BT, Vodafone, and InterDigital Communications.

References

[1] 3rd Generation Partnership Project (3GPP). (2007, Mar.). Study on facilitation of machine-to-machine communication in 3GPP systems. 3GPP Tech. Rep. 22.868, version 8.0.0 [Online]. Available: ftp://ftp.3gpp.org/Specs/archive/22_series/22.868/22868-800.zip

[2] 3rd Generation Partnership Project (3GPP). (2007, May). Feasibility study on remote management of USIM application on M2M equipment. 3GPP Tech. Rep. 33.812, unpublished draft version 1.4.0 [Online]. Available: ftp://ftp.3gpp.org/tsg_sa/WG3_Security/TSGS3_55_Shanghai/Docs/S3-091154.zip

[3] Y. Qian and N. Moayeri, "Design secure and application-oriented VANET," in *Proc. IEEE Vehicular Technology Conf.*, 2008, pp. 2794–2799.

[4] "TCG mobile reference architecture specification," TCG Mobile Trusted Module Specification, Version 1.0, Revision 5, 2008.

[5] A. Leicher, N. Kuntze, and A. U. Schmidt, "Implementation of a trusted ticket system," in *Proc. IFIP Int. Information Security Conf. SEC2009*, Boston, MA, to be published.

[6] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," in *Proc. 11th ACM Conf. Computer and Communications Security*, 2004, pp. 132–145.

[7] J. Camenisch, "Better privacy for trusted computing platforms," in *Proc. 9th European Symp. Research in Computer Security (ESORICS'04)*, 2004, pp. 73–88.

[8] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Commun. ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[9] V. Haldar, D. Chandra, and M. Franz, "Semantic remote attestation: A virtual machine directed approach to trusted computing," in *Proc. USENIX Virtual Machine Research and Technology Symp. (VM '04)*, 2004, pp. 29–41.

[10] A.-R. Sadeghi and Ch. Stüble, "Property-based attestation for computing platforms: Caring about properties, not mechanisms," in *Proc. 2004 Workshop on New Security Paradigms NSPW '04*, 2004, pp. 67–77.

[11] L. Chen, R. Landfermann, H. Löhr, M. Rohe, A.-R. Sadeghi, Ch. Stüble, and H. Görtz, "A protocol for property-based attestation," in *Proc. 1st ACM Workshop on Scalable Trusted Computing (STC '06)*, 2006, pp. 7–16.

[12] Trusted Computing Group (TCG). (2006). TNC architecture for interoperability. TCG Specification Version 1.3, Revision 6 [Online]. Available: http://www.trustedcomputinggroup.org/resources/tnc_architecture_for_interoperability_version_13

[13] Trusted Computing Group (TCG). (2008). TCG infrastructure working group, architecture—Part II—Integrity management. TCG Specification Version 1.0, Revision 1.0 [Online]. Available: http://www.trustedcomputinggroup.org/resources/infrastructure_work_group_architecture_part_ii__integrity_management_version_10

[14] A. U. Schmidt, A. Leicher, and I. Cha, "Scaling concepts between trust and enforcement," in *Trust Modeling and Management in Digital Environments: From Social Concept to System Development*, Z. Yan, Ed. Hershey, PA: IGI Global, to be published.

[15] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence, "Generic AAA architecture," Internet Engineering Task Force Network Working Group, Request for Comment (RFC) 2903, 2000.

[16] I. Cha, Y. Shah, A. U. Schmidt, A. Leicher, and M. Meyerstein, "Security and trust for M2M communications," presented at WWRF Meeting 22, Paris, France, 2009. ***VT***