|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Title\*:** | **Considerations on M2M security interworking with M2M Areas Networks, for M2M Release 2** | | |
| Submitted by: | Francois Ennesser ([francois.ennesser@gemalto.com](mailto:francois.ennesser@gemalto.com)) | | |
|  |  | | |
| from **Source**\*: | Gemalto | | |
| Submitted **To**\***:** | ETSI TC M2M | | |
|  | | | |
| Relevant WI(s), or deliverable(s): | Release 2 requirements and architecture  Interworking with M2M areas networks | | |
| Agenda Item: |  | | |
| Submission date**\***: | 2012-05-30 | | |
|  | | | |
| **Document for\*:** | Decision |  | 🡨 a decision is formally requested from the addressed (sub-)committee |
| Only one "**X**" | Discussion | **X** | 🡨 the contribution is expected to be presented and discussed, but no decision is formally requested |
|  | Information |  | 🡨 the contribution does not require discussion |
|  | | | |

**Decision/action requested** (Mandatory if Decision box is checked, optional otherwise)

*The present contribution provides information to consider for extending M2M Security Procedure to address D’ devices in M2M Release 2 Specifications. It considers M2M Area networks in their most general context (wireless, capillary, self-organizing communication of mobile devices...) and investigates interworking scenarios involving M2M Service Layer security to provide added value at the service layer. The need to extend M2M Service Layer security solutions to D’ devices is especially relevant for the purpose of offering End-to-end security services to M2M Applications. Several scenarios where security extensions addressing devices on M2M Areas Networks would enable M2M Service Providers to provide enhanced services to M2M Applications are highlighted. This contribution could serve as a basis to develop Release 2 Change Requests, once the committee agrees on directions.*

# In the rest of this document, the following formatting is used to facilitate the reading:

* Technical assumptions for the considered scenarios are underlined
* **Assessment of suitability (advantages/drawbacks of scenarios) are in bold**
* *Considerations of impact on TC M2M standardization activities are in italic.*

# **Rationale**

This contribution investigates possible synergies to achieve by bridging M2M core networks and M2M Area Networks in terms of security, especially in the context of mobile devices with limited communication range relaying their communications within MAN. We investigate a number of possible scenarios for this, which would enable M2M Service Providers to extend their service offer to M2M Applications. The focus is to enable “global” ad hoc communications whereby capillary devices will be able to relay M2M communications not only within their own capillary network, but also into and through other users capillary networks.

**Background on M2M Areas Networks**

M2M Area Networks are possibly highly dynamic networks that frequently change of topology as a result of devices mobility, with frequent devices joining and leaving, and possibly long distances between nodes. Not only M2M D’ devices but also M2M gateways (e.g., in vehicular networks) can be mobile devices.

In a multi-domain setting such as a vehicular networks, an M2M D’ device may not be able to directly reach the network of its administrative domain. It may connect through a network from another domain to which its own domain has some business relationship. Network access should be protected based on authentication procedures within such multi-domain infrastructure.

This contribution considers the general case where M2M Area networks may be capillary wireless communication networks, i.e. M2M D’ devices are limited in terms of communication range and may not be able to reach the gateway directly. They may rely on other devices to relay their messages, for instance their authentication request messages. Intermediary devices would relay such request messages toward the M2M gateway. After successful authentication, all cryptographic keying material needed for secure packet forwarding would be provided to devices to build link-layer security associations.

M2M devices may move separately or in bulk (e.g., devices attached to passengers in a bus). Re-authentication procedures of the bulk of devices may result in large communication overhead at the network access entity. A key challenge is to propose re-authentication and network access control procedures to handle such peak overhead. For instance, organizing the bulk of devices into groups should be supported to reduce management costs. The movement of the bulk of devices may be coupled with the movement of the gateway. In this case, delegation mechanisms could be supported to handle the re-authentication procedure of the bulk of devices.

**Background on Security**

The current TC M2M service bootstrap procedures serve the purpose of establishing initial shared secrets between devices/gateways and the Network Service Capability Layer (M2M service provider). The shared secrets defined in the initial run of the service bootstrap procedure may be used afterwards on a regular basis to derive shorter lived keys which are used to secure M2M communications.

The same need to define initial secrets arises also for D’ devices in M2M area networks located behind M2M gateways. But this has not yet been addressed by our Release 1 security architecture, which only addresses G and D devices directly connected to the Network domain.

Security bootstrap in M2M Area networks may target the establishment of pairwise keys or a group key among the communicating devices. Security bootstrap may also target the publication of a public key associated to each device to other devices. The public key is then used in conjunction with the private key of each device to establish secure shared key and/or shared group key through the authenticated key establishment mechanisms or to directly secure communication between devices using asymmetric cryptographic techniques. Pairing methods in M2M Area Networks can be classified in two categories: On the one hand self organized pairing methods which do not require the presence of a particular leader in the group, and on the other hand, methods relying upon the presence of a special group leader node, driving the pairing process. The latest are shown to be less demanding in terms of resources and computation on the D’ devices. We assume here that their specification is not a priori under the control of M2M Service Providers or in the scope of TC M2M.

## 1 Gateway involvement in Security Interworking with M2M Area Networks

In relation with TC M2M specifications, we can expect M2M Gateways to act as group leaders for MAN security pairing. A M2M gateway has generally less constraints in terms of processing power and energy than the capillary devices themselves. Furthermore it may be used to implement some type of possibly web based user interface that will make the administration of the devices easier for the user.

Three possible security configurations are investigated:

1. Gateway acting as a funnel for data communication originating from D’ devices. Each D’ device can connect independently to an infrastructure M2M network using its own identity.
2. Gateway acting as a data aggregator, connecting to an infrastructure M2M network with a single identity and relaying data to devices in the capillary network.
3. Gateway acting as a mediator connecting to an M2M infrastructure network with its own identity**,** in order to use on the MAN side security keys defined on the Network Domain side

To detail those 3 solutions we make the assumption that the security bootstrapping procedure on the M2M Area Network is leading to the definition of a group key.

When using a security bootstrapping methods described above based upon asymmetric cryptography and leading to a secure publication of devices public key, we will make the assumption that the public key of each device is used to securely transmit a group key shared by all peers belonging to the ad hoc network. This assumption is justified by the reduced computing power required by symmetric cryptography compared to asymmetric cryptography.

### 1.1 Gateway acting as a funnel to/from D’ devices on its MAN

This scenario is sketched on Figure 13. In this scenario, 2 distinct layers of security are involved:

* **Data communication in the capillary network are secured using the Kg key shared to all devices up to the gateway sink.**
* **Each D’ device number "i" in the MAN defines its own service key Ka[i] to secure its communication on the WAN side of the gateway.** When the last step involves an M2M Service Provider M2M service provider, this can be done either:
* With a hop by hop data protection scheme, where each segment of the data transmission from source to destination is protected with different keys, as in Release 1 TC M2M architecture.
* With end to end data encryption obtained with the help of an external authorization server (as described in M2M(12)19\_090 and subsequent CRs).
* Communications security may also be achieved via a peer to peer negotiation between one of the capillary devices with a remote peer, using previously established shared secrets.

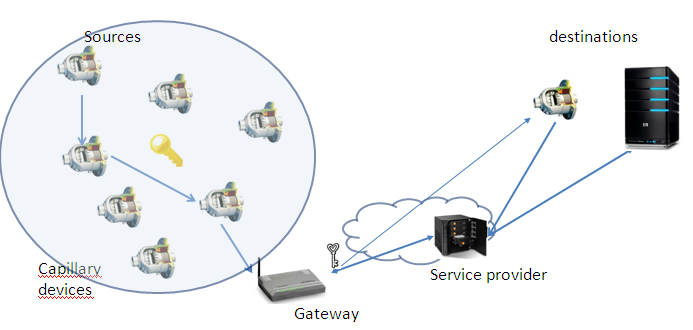


Figure 13: security scenarios involving different keys on the M2M core and on the MAN side

The advantage of the scenario is the **possibility for each device of the capillary network to generate traffic with its own identity and its own security**. First key Kg is used to relay data from the capillary node up to the sink (gateway). The second key Ka[i], negotiated at the capillary node [i] level is used to make sure that **the data from one node remains opaque to the other relaying nodes.**

**Such a scheme is therefore suitable when each node of the capillary network needs to secure its own communications with respects to its peers, while still using their data relaying capability.**

*Our analysis is that this scenario is neither explicitly forbidden nor currently supported by TC M2M specification, since the possibility for D’ devices to support the capability to negotiate their own keys through their gateway has not been considered.*

### 1.2 Gateway acting as a data aggregator (proxy)

Figure 13 is also suitable to describe this scenario. In this case, like in the previous scheme, Kg is only used in the MAN under the gateway, but only the M2M gateway, acting as an independent device, bootstraps the security on the M2M Core side using one of the methods described in the above paragraph. This results in the definition of a single **application Ka, used to encrypt data communications from/to the gateway.** The difference with the previous scheme is that **from the perspective of the M2M service provider, there is a single identity involved**. **The fact that the traffic is generated by multiple devices is opaque to the M2M service provider.**

The scenario described is very close to the proxy concept used in computers to achieve protocol translation. **The M2M gateway decodes data transmitted from / to D’ devices protected with Kg, and re-encodes it with the Ka key prior to transmission to the M2M/service provider or directly to a remote peer, possibly using a different transmission protocol.**

*This protocol translation could be achieved at the application layer. In Annex B we detail a possible implementation using COAP as a transmission protocol on the MAN side and investigate the impact of other existing specifications.*

The advantage of this scenario lies in the fact that **it reduces the computing burden for the D’ devices which may be constrained both in energy and computing power, by avoiding a dual encryption scheme.** **Each D’ device can also handle its own communications with whatever source or destination address it may chose.** However, all the D’ devices share the M2M gateway identity.

**Such a scheme is suitable when the D’ devices do not have the need to protect their data with respect to their peer.** Data exchanged in the MAN via relaying is readable by all nodes. **From the outside world, the gateway appears with a single identity, hidding to the M2M Service provider the details about the D’ devices involved in the aggregated traffic generation.**

**A disadvantage lie in the fact that data protection is piecewise, and the gateway needs to be trusted in order to achieve suitable security.** This leads to a couple of sub-cases:

#### 1.2.1 Data aggregation in private gateway

T**he M2M gateway is “private” when the Gateway and all D’ devices conneted to it belong to the same application, so that the Gateway doesn’t carry any other application traffic.** In this sense it is like a closed-mode femtocell or (home/enterprise) WLAN access point. *This seems to be an implicit assumption in Release 1 TC M2M specifications, i.e. the gateway is likely to be owned the application provider.*

In such cases, the threats are mostly limited to external parties attempting to corrupt or hijack the gateway or other devices in the MAN, e.g. by logical attacks exploiting vulnerabilities in the gateway software. There are some physical attack scenarios as well (if, for instance, the gateway is in an accessible outdoor location), but **the gateway owner has an incentive to protect against both logical and physical attacks.**

#### 1.2.2 Data aggregation in public gateway

When a particular MAN communication technology prevails across multiple application providers, it could be of interest for the M2M service providers to deploy its own **gateways shared between multiple applications,** or to leverage on **existing gateways already** **carrying other traffic than the one that needs to be aggregated for a particular application.** In this sense it is more like an open-mode femotocell or public WLAN hotspot.

Though such M2M gateways may still be owned by the party that owns the device, it may also be owned (or subsidized) by the M2M service provider, or access network provider. Again **this scenario gives the gateway owner an incentive to protect against logical and physical attacks, but not as strong as if it was the owner of aggregated traffic.**

In such scenarios **the gateway should preferably be partitioned / segregated so that each aggregated private traffic and any potential public traffic are all handled separately.** Protection against logical attacks arising from open access is one reason to segregate traffic streams in a “public” gateway case.

*Such scenarios, of obvious interest for M2M Service Providers, do not seem precluded in TC M2M Release 1 specifications, but explicit mentions of requirements applying to Gateways in such cases would be useful to acknowledge their consideration.*

### 1.3 Gateway acting as a mediator between the MAN and the M2M Core

A variant of the above scenario, shown in figure 14, consists in linking M2M Core and MAN security. In this case, the M2M gateway participates as described in section ‎2.2 in the pairing definition in the MAN and performs as well a M2M security bootstrap. The M2M gateway then communicates the Ka key obtained from the M2M Core to each D’ device to be used end-to-end as a group key. Kg resulting from the pairing defined on the MAN side is used to safely transmit the obtained service key Ka to each of the D’ devices. Each D’ device then use only this Ka key to secure their communication ed-to-end.

Figure 14: security scenario involving the same application key on the MAN and on the WAN side

**The advantages of this scheme lie in the fact that devices only have to implement a single encryption layer to secure the whole transmission path. It removes the need for data re-encryption in the M2M gateway.** However the size of the Ka key and the cryptographic algorithms used must be compatible with the computing power available in the D’ devices.

*Though this scheme has not been considered in current M2M Specifications, we would advise to specify the corresponding M2M gateway behaviour within the context of the security framework extension to address end-to-end security. This would in effect extend the scope of end-to-end applicative security to D’ devices, through dIa, while it is otherwise limited to D/G entities directly connected over mId.*

Leveraging on this model would enable an M2M Service Provider to aggregate M2M Areas networks belonging to distinct owners, enabling D’ devices.of one application to use communication capabilities from devices or gateways of other applications. This may be especially valuable for mobile D’ devices using limited range wireless communication relying on capillary communication between neighboring D’ devices to reach their M2M Gateway. Thare are many use cases where such deployments models are required, such as fire sensors deployed in forests.

## 2 Infrastructure assisted bootstrap in and between M2M Area networks

This section identifies a number of scenarios that create a synergy between the M2M Core and M2M Area Networks, in order to enable “global” ad hoc communication. We especially consider the case where the MAN is a capillary network, in which D’ devices belonging to one owner are able to channel their communication via other D’ devices possibly belonging to another owner, in order to reach their gateway and achieve global ad hoc coverage at lower costs and when mobility is supported.

Three scenarios are considered along that line:

1. **Two D’ devices belonging to 2 distinct MAN communicating together via the M2M core (no proximity communications)**
2. **Single D’ device having its data relayed by a guest capillary network, after a phase of security bootstrap in its own network**
3. **Bridging two D’ devices belonging to different users, so that capillary D’ devices of the first network have their communications relayed possibly by nodes of the second network, in effect aggregating the capillary MAN of each application to enhance coverage for end devices.**

### 2.1 Single D’ device connecting to a guest M2M Area network

This scenario is outlined on Figure 16. The capillary D’ device 1 represented on this figure has already performed a pairing process in its home MAN. The M2M gateway in this home MAN acted as a group leader in the pairing process and obtained credentials as a M2M gateway with an M2M service provider. The MAN 2 belongs to another user affiliated to the same M2M service provider.

As a result of this scenario, when placed in a mobility situation, Device 1 becomes able to route its data via the self organized MAN 2. This of course requires the capability to incrementally add and revoke devices in the MAN: Security requirements for this capability are considered in Annex C.

The solution described below relies on an authorization architecture such as proposed to address End-to-End security in M2M Release 2. The scenario, resulting in D’ device 1 getting access to MAN 2, can be summarized as follows:

1. the gateway of MAN 1 is authenticated by a remote authorization server and obtains a signed “delegation” electronic token
2. The gateway of MAN 1 then provides device 1 with a signed electronic token enabling it to be authenticated with foreign MANs.
3. Device 1, when in a mobility situation, want to connect to foreign MAN 2 and presents its electronic token which is verified (possibly resorting to the authorization server)
4. Device 1 is granted access to MAN 2 and is provided with the group key needed to use relaying in this network



Figure 16: Single device connecting to a foreign MAN

Figure 17 describes a workflow suitable to enable this functionality. The flow of operations, leading D’ device 1 originally part of its own MAN (capillary network 1) to be able to communicate when placed in a mobility situation through capillary network 2, may be decomposed in 2 stages:

1 Operations occurring while device 1 is still within capillary network 1

2 Operations occurring when device 1 is in a mobility situation

1. **When device 1 is within capillary network 1** :
   1. The gateway of the capillary network 1 performs a bootstrap operation with the M2M service provider using one of the TC M2M security bootstrap methods, and this bootstrapping process results in the definition of a shared root key : Kr
   2. The gateway generates a pair of asymmetric keys, creates a certificate signing request and ultimately obtains a certificate from the M2M service provider, e.g. using GBA or TLS bootstrapping specified by TC M2M.
   3. The device takes part in the pairing process at the MAN level. The gateway plays the role of the group leader in the pairing process. The pairing method used in this case is assumed to enable the publication of devices public keys. At the end of this process the gateway owns a copy of the device public key and creates a signed certificate using the certificate it obtained in step b. The public key of the device may then be used to distribute a shared group key enabling the capillary D’ devices to relay their communications in the capillary MAN.
2. **When the device is in a mobility situation, outside its home capillary network**
   1. D' Device 1 then may want to connect to a host capillary network. To do so, it performs a pairing operation with any of the devices of the host network, presenting its signed public key to the host D’ device. At this point, 3 scenarios may be envisaged:
      1. The host device may be able to check the signature (and verify the PKI chain) locally and decide on granting access to the guest device. This solution will make a mutual authentication difficult, as the host device may not have the proper credentials to be successfully authenticated by the guest device.
      2. The host device may resort to the local gateway for granting access to the guest device. The decision is taken locally at the gateway level. Mutual authentication is possible if the gateway presents its own signed certificate obtained from the M2M service provider.
      3. The host device resorts to the gateway to verify the signature locally, and then communicates with a remote authorization server, which will ultimately grant or deny access. Here again, the gateway may be authenticated by presenting its own certificate.
   2. Once access is granted using one of these 3 methods, the guest device obtains a copy of the shared group key used for relayed communications in the capillary network and is then able to relay communication via the other devices in the MAN. Another alternative could be for the guest to obtain a guest key with limited privileges, and simply enabling data relaying.



Figure 17: workflow for single device connecting to a foreign capillary network

**The workflow described above can be generalized to describe a more general security architecture, featuring clearly defined policy decision points (PDP) and policy enforcement points (PEP).** Furthermore, the workflow defined here does not take into account the definition of security policies. Such *policies could be envisaged at several levels:*

* *At the level of the guest network:* 
  + *Policy defining the acceptance of guest request ( for exemple, a gateway may be accepting guest requests only from 5 to 7PM)*
  + *Policy definition conditions for granting access (i.e. who can be allowed as a guest)*
* *At the level of the connecting guest:*
  + *Policy defining connection attempts (for example, only connect to networks affiliated to a specific M2M service provider)*

### 2.2 Communications between D’ devices via the M2M Core

This assumes that the gateway plays the role of a mediator between the MAN and the M2M Core, in order to enable D’ devices to communicate securely with the M2M Core without the need for data reencryption.

This also relies on an authorization server offering a “key ” distribution service, as proposed in M2M(12)20\_111.

We show here how these 2 concepts can be combined in order to enable data relaying between devices belonging to distinct capillary networks.

The overall architecture used for the discussion is shown Figure 18. This shows 2 capillary networks. The devices in each of those MAN relay each other data in order to communicate with the outside world through their own gateway. However relaying initially occurs only within each capillary network, reflecting the fact that the two networks are disjoints (possibly belonging to 2 distinct owners)



Figure 18: bridging capillary networks via M2M infrastructure network

Each of the gateways creates a working session with an M2M service provider and authenticates with an authorization server in order to obtain a signed “delegation” electronic token.

Gateway 1 then provides to D’ device 1 a signed electronic token enabling it to be authenticated with the authorization server, via a verification of the PKI chain. Gateway 2 performs the same operation with device 2.

Device 1 and device 2 both connect to the authorization server and are authenticated using the signed electronic token delivered by their respective gateways. Their communications rights are verified, and they receive a secret key Ke that will allow them to communicate together.

**This scheme presents the advantage of enabling one or several D’ devices of MAN 1 to communicate privately with one or more devices of MAN 2. It could be easily extended to create multiple simultaneous communications, all protected by different secrets.**

**It is suitable for use cases where each D’ device should be able to establish its own private communications. Another advantage is that the outside world only sees the identity of the gateways, and the details of the MANs remain opaque.**

A drawback of this scheme is that **each D’ device should perform a dual encryption**:

* One encryption using the capillary network group key, enabling capillary devices communication to be relayed by their peers.
* One encryption of the payloads for end to end data encryption via the M2M Core. This is the price to pay in order to be able to secure capillary communications at the device level.

For use cases that do not require confidentiality at the device level, it is possible for the gateway to act as a proxy. Let’s detail how this would work:

* Each of the gateways shown on the figure creates a working session with an M2M service provider and authenticates with an authorization server in order to obtain a secret key Ke enabling them to securely communicate together.
* Each of the gateways participate to the security pairing in their own network, and this results in the definition of the keys Kg and Kc enabling respectively secured relaying in capillary networks 1 and 2.
* Device 1 sends its data to gateway 1 protected with the key Kg. Gateway 1 relays the data to gateway 2, protected by the key Ke. Finally gateway2 relays the data to device 2 using the key Kc ( cf Figure 18).

### 2.3 Aggregation of capillary networks

As well as bridging between D’ devices at the application level, it would also be desirable to bridge between geographically nearby D’ devices at the network level. This allows Device 1 and Device 2 to “cross-connect”, without routing all their communications through gateways 1 and 2 and the access network. The architecture is shown below.



Figure 19: Bridging capillary networks via M2M infrastructure network

One solution here is to use the security described in Figure 19, except that the key distributed to Device 1 and Device 2 is used as a pre-shared key for authenticating (and authorizing) a network-layer pairing between them. Also, the “authorization server” here may well be hosted by the M2M Service provider, because **the M2M SP is likely to have the information about the geography, topology and wireless technology used by overlapping capillary networks, and can identify devices which could potentially cross-connect.** Further, the cross-connection may not be for a specific application purpose (it might be a general routing optimization affecting several applications). **This solution will work in a “planned” cross-link scenario:** **the M2M service provider identifies devices which may cross-link, and instructs them to do so.**

An alternative solution is an **“opportunistic” cross-link scenario, whereby Device 1 just detects that it is in range of Device 2 and attempts to join Capillary Network 2** (while never actually leaving Capillary Network 1). The architecture that would work here is the **“roaming” solution described in 20**. Effectively Device 1 “roams” onto Capillary Network 2. Communication with the authorization server after connection (to confirm that the opportunistic connection was acceptable) is still needed.

However, if we now make the assumption that device 1 and device 2 cannot communicate directly together as shown on Figure 20, but rather need to have their data relayed by other nodes of the capillary network, then we **need to perform a full aggregation of the 2 capillary networks.**



Figure 20: Bridging capillary network through proximity communications

One solution is to reuse the scheme described in section ‎2.2 and have each gateway distribute in their own capillary network the application key Ke used for end to end data encryption, as illustrated in Figure 21.This would result in **the key Ke being used both to protect end to end data transmission and proximity communication.**

The drawback of this method however is that **all devices of the two capillary network may have access to all data transmitted, which may be a problem when the two networks are owned by distinct users.**

Figure 21: Linking WAN and capillary security

### Annex A: Comparison of Group Device Pairing Protocols

compares the group device pairing protocols (outlined in M2M(12)19\_091) from number of broadcast messages, OoB message size, required cryptographic primitives (H: hash function, UH: universal hash function, MAC: message authentication code, D: digest functions, C: cryptographic commitment scheme, XOR: exclusive-or operation), communication and computation cost. In this table, the adversary success probability is limited to and so the out-of-band message is b-bits for all the protocols except group MANA protocol which is 2b-bits.

In computing communication cost we assumed that commitment value C is a fixed size value |C| and decommitment value D is all the committed values in addition to a randomizer K. We used |ID| for identifiers, |PK| for public keys, |K| for long keys (e.g. 160 bits), and also commitments’ randomizers, |R| for short random number (e.g. 32 bits), |H| for hash functions, and |MAC| for MAC functions. To give a better comparison, the table also shows the communication cost without considering public keys and identifiers (which are common for all the protocols) using W for 160-bits average size of K, H, C, and MAC and W/5 for 32 bits R.

**Comparison result indicates that group device pairing mechanisms with trusted leader have clear advantage over symmetrised protocols from communication and computation cost perspectives.** In relation with TC M2M specifications, we can expect M2M gateways to act as trusted leaders. Among the protocols with trusted leader, group numeric comparison protocol and MC-GDP protocol are not efficient respectively from communication over wireless and OoB channel. **Among the HCBK and group MANA II, we recommend using HCBK protocol as it requires less user effort over OoB channel and its inefficiency in sending 2W-bit instead of W-bit is acceptable as it is only for the gateway which is usually a powerful device.**

Table : Group device pairing protocols – communication and computation cost

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol** | # Broadcast Messages   per device | **OoB Message Size (bits)** | **Required Cryptographic Primitives** | Computation Cost per Device | **Total Communication Cost** | **Communication Cost without public keys and identifiers** | **Comments** |
| Group Numeric Comparison | 2  (L: 1) | b | CR hash function | N\*H | N[|ID|+|PK|+|K|]+(N-1)|H| | (2N-1)W | * with trusted leader |
| Group MANA II | 1  (L: 2) | 2b + 1  (1 bit for Ack. signal) | CR hash function + universal hash function | 1\*H +1\*UH | N[|ID|+|PK|]+|K| | W | * with trusted leader |
| MC-GDP | 1  (L: 2) | (N-1)b+ 2  (2 bits for Ack. signal) | message authentication code function | 1\*MAC | N[|ID|+|PK|]+|K|+ |MAC| | 2W | * with trusted leader |
| HCBK | 1  (L: 3) | b + 1  (1 bit for Ack. signal) | CR hash function + digest function | 1\*H + 1\*D | N[|ID|+|PK|]+|K|+|H| | 2W | * with trusted leader |
| GAP | 4 | b | cryptographic commitment + XOR operation | 1\*H + N\*C +  (N-1)\*XOR | N[|ID|+|PK|+|C|+ 2|K|+|R|] | 3NW+ NW/5 |  |
| SHCBK | 2 | B | CR hash function + digest function | N\*H + 1\*D | N[|ID|+|PK|+|K|+|H|] | 2NW |  |
| SAS-GMA | 2 | B | cryptographic commitment + universal hash | N\*C + 1\*UH | N[|ID|+|PK| +|C|+2|K|] | 3NW | * Provable Secure |

There are then several ways in which group keys established in MAN using the above algorithms can be used to secure communications within the capillary networks. This is driven by the answers to the primary questions below, which depend mainly on specific MAN technology and applicative requirements:

* *Should the group key be used directly for communications, or should it be used to negotiate link keys (between pairs of devices), or sub-group keys (for nearby devices)?*
* *Do the MAN communications involve integrity protection keys as well as encryption keys? Or is a hybrid mode (authenticated encryption) used?*
* *What crypto algorithms should be used, and are they block cipher, stream cipher or both?*
* *Are there protocols to negotiate the use of a particular set of crypto algorithms and parameter lengths (key sizes, MAC sizes, length of time and data until next negotiation etc.)? How are these protocols protected against downward negotiation or rollback?*
* *Does the implementation require additional features for security such as anti-replay counters, or random Nonces, or time synchronization between devices?*
* *If random values are needed (for Nonces or key derivations) then what mechanisms do the devices provide to generate these random values? Can they use a PRF for instance? What sort of PRF?*

These are all sensitive issues when designing secure communication protocols especially in wireless environments, but we assume that their answer depends on specific technologies and is therefore out of scope of TC M2M.

#### A.1 The example of OMA CPNS

The OMA CPNS (Converged Personal Network Service) enabler specification, released in May 2011 by OMA, defines CPNS enabler as a framework which provides a method for interconnection of various devices in Personal Networks (PN) to the WAN. Internet services and applications are available through a dedicated (gateway) device that is usually a mobile phone. The gateway device is called Personal Network Gateway (PN GW). The PN can be a capillary network (Bluetooth or WiFi). Again, protocol translation is required by the Gateway. CPNS Enabler 1.0 provides the means to ensure security in CPNS including authentication, authorization, data integrity and data confidentiality.

CPNS Enabler 1.0 has four security mechanisms:

1. Authentication/authorization of CPNS Users/CPNS Entities by Entity User Key
2. Protection of data integrity and confidentiality.
3. Security-key (Entity User Key and Group Key) management.
4. Secure content/service sharing inside Service Groups by Group Key

The security mechanisms specified by CPNS are applicable to some high end D’ devices for achieving pairing. This mechanism is known as **Gateway Assisted Pairing**. They are **probably not suitable for very low end devices**. Also CPNS does not curently consider relaying between PNE’s.

## ANNEX B: Implementation of protocol translation using COAP

The scenarios described in section 1 often involve the implementation of a protocol translation scheme, when the M2M gateway is funneling all D’ devices communications and needs therefore to associate each response received to previous requests from the devices.

The protocol translation operation described here require IP connectivity to the M2M Core for the gateway, but not necessarily for the D’ devices. We describe here how the protocol translation could be achieved when CoAP is used as a transport protocol over dIa.

In CoAP, optional (or default) request and response information such as the URI and payload content-type are carried as CoAP options. A “**Token**” Option is used to match responses to requests as shown on Figure ‎15.

This “Token” option could be used as a basis to perform address translation pretty much as the originating port number is being used in traditional NAT solution to route back to the right device the response to a previous request.

This solution would involve to use a lookup table maintained by the gateway, mapping Token values to capillary devices originating address.

It should be noted that the size of the Token option fields may be comprised between 1 and 8 bytes. *In implementations with large Token values, using Token option to do address translation may be unfeasible due to the size of the lookup table required.*

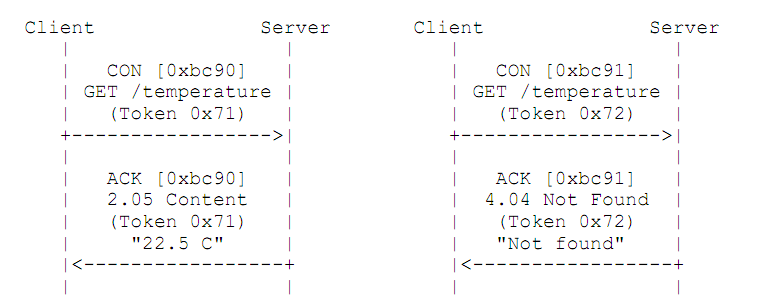


Figure 15: Token Option may be used in CoAP to match request and responses

#### B.1 IP/IEEE address translation gateway

IP/IEEE address translation at the gateway level is another option to allow D’ devices in the MAN without an IP stack implementation to be visible to the IP world. For example, a TCP socket procedure on M2M gateways may be used to connect to a server on the Internet and map D’ devices from its MAN into port-based connections.

#### B.2 6LoWPAN interoperability gateway

In 6LoWPAN, a gateway may be responsible for interoperability between 6LoWPAN and external IPv6 networks. With IP address / 6LoWPAN address translation capability, 16-bit short address end devices in the 6LoWPAN network are reachable by application/security servers. The M2M gateway may have internal and external device address mapping tables for this translation:

* Internal Device Address Mapping Table consists of 64-bit interface identifier (IID) and 16-bit short address. This table must contain the mapping information of all devices in the 6LoWPAN. The maximum size of the mapping table is 216 entries.
* External Device Address Mapping Table consists of 128-bit IPv6 address, 16-bit short address and ET (Expiration Time).

### ANNEX C: Incremental addition and revocation in M2M Areas networks

Incremental addition and revocation of devices in a group is an important issue especially in the context of capillary networks. Issuers of D’ devices should be able to revoke any of them, without tearing down the whole network, in a way that revoked device(s) cannot access the future communications (forward secrecy in group communication). This is also true for adding new D’ devices to M2M Area network, which should be done with minimum user effort in a way that a new device cannot access previous communications (backward secrecy in group communication).

The gateway plays a central role in the MAN’s key management. Regarding the pairing process on M2M Area Network, we assume that the gateway has authenticated public keys of all the paired D’ devices connected to it and securely distributed a group key among them, using methods such as those considered in Annex A.

* To revoke a paired D’ device, the gateway has to remove the public key of revoked node(s) from its list and distribute a new group key encrypted by the public key of the remaining D’ devices.
* To add a new D’ device to a MAN, the gateway has to securely pair with the new node, authenticates its public key and gives it an encrypted copy of the current group key. In addition, the group key needs to be updated periodically to ensure backward secrecy.

For pairwise pairing between the M2M gateway and new D’ device the following are possible:

* using well-tailored pairwise pairing device pairing protocols
* or applying the current group device pairing mechanisms.