sDOMO – A Simple Communication Protocol for Home Automation and Robotic Systems

Marcel-Titus Marginean
Towson University
Towson, Maryland, USA
mtm@mezonix.com

Chao Lu
Towson University
Towson, Maryland, USA
clu@towson.edu

Abstract— **Today's market for domotic systems is dominated by proprietary communication protocols incompatible with each other. This not only prevents engineers from building highly integrated home automation and domestic robotic systems, but keeps prices artificially high due to a lack of competition in the market since the customers have to restrict their choices to components compatible with the system already deployed. In this paper, we propose a communication protocol for fully integrated domotic systems running on top of house network, protocol which is simple enough to accommodate as peers micro-controller based sensors but scalable enough to handle a house wide distributed computer vision system for domestic robot assisted living. Based on a house centered philosophy, the protocol is putting emphasis on privacy and protection of the residents.**

Keywords—domotics; communication protocols;

I. INTRODUCTION

In our previous papers [1, 10] we introduced a distributed computer system architecture for domestic robots operations assisted by both external (house mounted) and local (robot) computer vision. One of the obstacles encountered during development has been the lack of a suitable communication protocol on top of which to implement the proposed architecture and this provided us with the opportunity to propose a new communication protocol for domotic systems which attempts to solve some major open problems in the field. Today's market for domotic systems is dominated by proprietary communication protocols incompatible with each other, without publicly available documentation or bounded by licensing fees and/or non-disclosure agreements. Besides preventing multiple vendor integration the current situation also hurts market competition which is required to bring the prices down.

To address the interoperability problem, a number of authors and open source enthusiasts proposed various solutions of the problem. However in the effort to integrate the existing devices many of the proposals from the members of the community rely on web based SOA (Service Oriented Architecture) using SOAP/HTTP and XML messages therefore paying a price in efficiency, bandwidth and CPU consumption in order to gain interoperability. Another problem is the fact that some Internet companies which are making most of their revenues from collecting and selling personal information are entering into this new field. The model for home automation promoted by these vendors are based on the over-hyped concept of "Internet of Things" (IoT) where each device installed in a user's house has a direct connection with the vendor's network; the residents of the house have to use Web services and applications provided by the vendor in order to control their own domestic environment. This has the potential to create a very dangerous precedent of domotic equipment spying on the residents in the privacy of their own homes and they attempt to justify this total lack of privacy as a necessary step toward achieving the intelligent house of the future. We reject this idea!

In this paper we introduce the Simple Domotic Protocol (sDOMO) which employs binary packed messages sent over raw UDP for efficient usage of network bandwidth and CPU resources. The binary protocol is efficient while performing frequent operations associated with core communication, allowing small, micro-controller based devices to operate as peers in the home automation network. In the same time, the power of expression and extensibility of XML is employed at the higher powered House Hub, House Intelligence Unit, Administrative Console and by the developer tools which are used to generate code to parse/build the protocol messages. Our vision regards the whole intelligent home as an integrated domotic system working together to keep the inhabitants comfortable, the robot being just a mobile component of the smart house of the future. An integral part of our vision is the concept of a self-contained domotic system which is able to solve most of the problems without any reliance on the internet or vendors sites. However, when such assistance is required, the accesses are performed via an Internet Gateway able to protect security and privacy of the domestic network without disclosing private information; "what happens in the house stays in the house" is the motto of the presented architecture which is introduced in this paper.

A relatively similar architecture with ours having closely related security goals has been presented in [2], however the reliance on public key infrastructure will not permit smaller devices to connect directly with the rest of the network, therefore the system must employ Room Bridge acting as protocol adapters. While the bridging architecture is also supported by sDOMO for devices that are unable to speak the protocol, our work reduced the protocol requirements to the basics, allowing much smaller devices to be part of the network directly. In [8] a microcontroller based module has been presented but it relies on external serial-to-lan converter and no overall domotic network architecture has been attempted. An IP based home automation system has been presented in [9] but the reliance on SSH excluded microcontroller based devices from working as peers. To solve the interoperability problem between otherwise incompatible platforms in [3] the idea of a SOAP based middle-ware has been proposed. A similar SOAP based middle-ware has been presented in [7]. The high overhead and verbosity associated with SOAP however would require that any small sensor or device to depend on a more powerful adapter. A similar idea with our Home Intelligence Unit (HIU) has been presented in [4] by Kao and Yuan where

they developed a more elaborated and complex set of meta-rules than our HIU. We may be looking more in depth at their ideas for our future development of HIU. The idea of a framework for developing domotic systems by following a model driven approach has been introduced in [5], while [6] presented a state of the art report on home automation technologies.

The paper is organized as follows: In section 2 an overview of the system architecture is presented and the main components of the system are introduced; section 3 highlights the core ideas of the communication protocol while section 4 looks at the security and privacy involved into the domotic system. In section 5 the configuration files and additional subsystems are presented and the paper ends with the conclusion, results and future lines of work.

II. SYSTEM ARCHITECTURE

sDOMO protocol proposes a hybrid communication structure with a House Hub acting both as a central registry for all the devices and also providing reliable and secure communication between devices; and optional direct device to device communication channels. Besides the home automation devices and the House Hub, the system contains a few special software components like House Intelligence Unit (HIU), Internet Gateway and Administrative Console used to control or complete the functionality of the system.

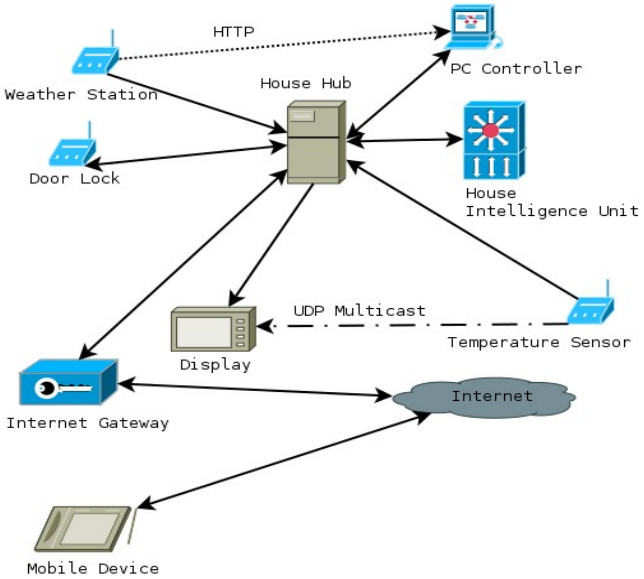

Fig. 1. Hybrid Communication Architecture

A Device is the generic name for a node in a domotic network. It can be an embedded device like a microcontroller based thermometer having Ethernet connectivity, a serial device speaking sDOMO over SLIP or PPP and connected to a serial-to-lan converter or a program running on a laptop or tablet used to display data or to manage the sDOMO network. Bluetooth and ZigBee are also potential candidates to be considered in the future. The network associated to a particular house is called a Domain and is defined by a Domain Name. A Device is identified by the Device Unique ID (DUID) which is a 20 bytes alphanumeric string constructed based on a set of rules to insure uniqueness among vendors and/or hobbyists. Once a Device joins a Domotic Network by being accepted to a Hub, the Hub issues a Device Session ID (DID) and a Session Key is used to authenticate messages and eventually encrypt the communication between the Device and Hub. The type of devices is defined by the Spec ID. This is a string of up to 128 characters that is define as a URL from where the Spec File can be retrieved. For example if a Device advertises a Spec ID of: *mezonix.com/sDOMO/devs/dev1* that means that a XML file defining the device can be downloaded either from: *mezonix.com/sDOMO/devs/dev1/device.spec* or from *www.mezonix.com/sDOMO/devs/dev1/device.spec* either by HTTP or HTTPS.

A Spec File is an XML file describing the device, linking to user and developer documentation, defining connection requirements, and the interfaces of the software objects implemented in the device. For example an indoor video camera can specify that it transmits sensitive data therefore the Hub may decide to require encryption. What types of encryption are available for a given device, their strength, and order of preference is also specified in the Spec file. For privacy enforcement a Spec File also contains a list of the interfaces that state this device must connect as a client in order to perform its tasks. From the software architecture point of view a Device is composed of one of more (up to 255) Objects identified by a contiguous number between 0 to NoOfObjects. Each Object implements an Interface whose Definition File if referred by the Spec File. The Interface Definition File (IDF) is another XML file defining the list of all Messages understood and/or sent by a given Object, the way they can be associated to provide method calls and the events that triggers them. The Interface Definition file is used by the developer tools to generate code for a Client that connects to an Object or for the Stub that implement the respective functionality. It is also used by the Hub for privacy enforcement and by the House Intelligence Unit for alerts and adaptive automation.

Each Device has a Device Main Key which is used by the Hub to encrypt the Session Key when it is delivered to the Device. A Hub that will accept a particular device must know the Device Main Key and its DUID, which is usually uploaded by an administrative application scanning a QRCode or RFID tag attached to the device or typed in by the user.

Two sDOMO Devices can communicate either via Messages routed by the Hub either talking directly to each other or bypassing the Hub in what is called Streaming. Messages are blocks of data up to 4GB in size (subject to Hub/Device memory limitations) transmitted between Devices. Messages are carried as a payload in Packets. A sDOMO Packet is sent into a single UDP Datagram over the local LAN. Packets are sent between a Device and the Hub only, they are the backbone of connectivity and they can carry fragments of Messages as their payload. Messages are routed by the Hub from the source to destination Device. There are two different types of Messages: Notifications and Direct Messages. A Notification is a sDOMO message sent by a device and received by any number of devices that were subscribed to it prior to the moment it was sent. A Direct Message is a one to one communication sent by an Object, part of a Device specifically towards another Object of a connected Device. Besides carrying Messages and acknowledgments for them there are various packets for device discovery, configuration, connection maintenance etc. The full definition of the packets structure

and a C++ library for working with them is provided on the companion website for this paper.

A Virtual Device is a set of services provided by the Hub to the connected devices via Direct Messages and Notifications. In order to facilitate the process of discovery the Virtual Devices have well known DID's allocated into a reserved range from 1 to 999 in which only Virtual Devices and Devices with Special Duties are allocated. The Virtual Device No. 1 have the object #0 designated as the DeviceManagerInterface allowing other devices to list the connected devices and their objects and get extended information about them. The AdministratorServices object allows for a console program to upload devices keys and edit their parameters. The AnonymousServices allows a device with level of trust 0 (without a recognizable Main Key) to log in as a program with special access or administrative rights using a user/password credentials pair or presenting a PKI Certificate signed with an authoritative Private Key. The House Intelligence Unit (HIU) is a special device acting as a meta-rules processor and is responsible for handling out of ordinary conditions, issuing alerts and perform complex automation tasks coordinating multiple devices based on scripts.

III. DATA COMMUNICATION

A. Overview

The preferred method of communication envisioned by our design is for devices to communicate with each other by Messages routed via the House Hub. The Hub receives packets, validates their authenticity, decrypts data payloads if necessary, assembles their payload into messages and delivers them to their destination device. While delivering messages, the Hub will enforce security and privacy rules on behalf of the devices, allowing small devices to benefit from the same level of protection as larger devices.

To support direct communications sDOMO allows for each device to advertise to other components in the network their own external/proprietary interfaces in this case sDOMO hub acting exclusively as a rendezvous server, allowing other devices to discover which peers are on the network and how to talk with them. This feature is implemented in order to allow vendors a seemly transition from their current protocols to sDOMO, or to support special high throughput applications like High Definition Video Streaming for which the House Hub would just introduce unnecessary delays. The protocol allows for multiple computers running Hubs for a single Domain however from the software point of view this is transparent to the devices.

The sDOMO architecture defines a house-centered self sufficient network model. Internet connectivity is allowed exclusively via an Internet Gateway which isolates the sDOMO network from the cloud. The Internet Gateway provides encrypted communication with the outside world and require authentication of the users requiring access. The access to information is controlled via Access Control Lists indexed in the user id. In order to allow remote sensors and sDOMO Devices, the Internet Gateway provides sDOMO tunneling toward a specified IP address. Supplementary, regular web application can run in the Gateway allowing controlled access via a regular browser to specific functionality.

There is no restriction for a sDOMO device to have other interfaces incompatible with sDOMO. For example a Power Meter can have a totally proprietary interface to be accessible by the power company while also exporting a sDOMO device in order to inform the household monitoring software about their energy consumption. The sDOMO specifications are responsible for defining a house-centrist architecture and to not interfere with proprietary interfaces of leased devices..

B. Discovery and Configuration

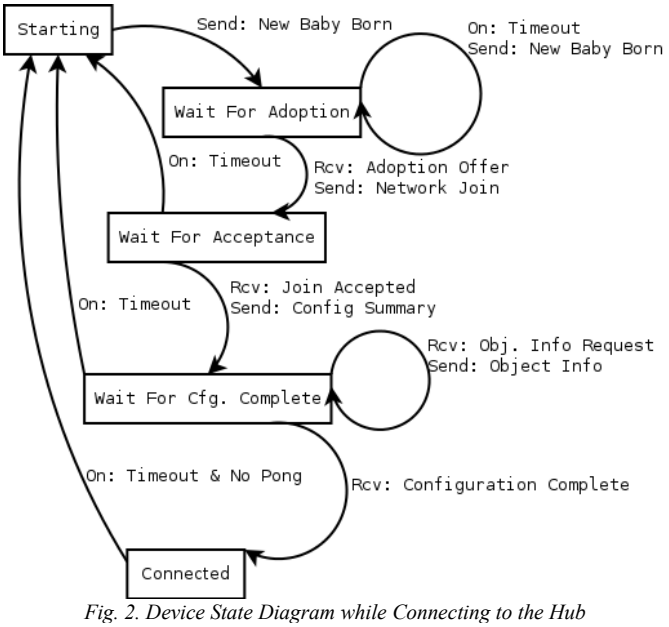

Fig. 2. Device State Diagram while Connecting to the Hub

A Device starting up will use Multicast to the proposed sDOMO Local Mcast destination (224.68.12.8 port 1881) sending a "NewBabyBorn" Packet containing it's DUID, Spec ID and previous Domain and Given Name if any. A Hub that matchs the Domain (if present) and wants that Device to join its network will send an "AddoptionOffer" Packet specifying an adoption bid between 0 and 255. The Hub that sends the highest bid will be selected by the Device to join by sending a "NetworkJoin" packet. The Hub accepting a Device replies with a JoinAcceptedPack which contains the Devices new DID and Session Key. The reception of JoinAcceptedPack triggers the Device to send the ConfigurationSummaryPack containing some configuration parameters that can overwrite the default values specified in the Device Spec File. For each Object of the new device the Hub query for that object information and the Device replies with an ObjectInfoPack. Once all the Objects have been configured the Hub sends a Configuration Complete Packet which switches the Device to full operational mode. From this moment the device can start communicating with other devices via Notifications or Direct Messages. Having a Session Key received the device signs all outgoing packets and validates the signature of on any incoming packets.

A Device whose Device Main Key is unknown by the Hub will receive an adoption offer with a bid of 0. A regular device will ignore any bid with value of 0. However an Administrative Console program or other GUI based device that can prompt the user to type in a username and password to join the hub at trust level 0 and use Anonymous Services to upload it's Main Key encrypted with the user's credentials. Such GUI based application are then used to scan and upload to the Hub the DUID / Key pair for any new device that will be connected to the network.

A second type of Main Key upload procedure was developed for the usage of sDOMO as a Robotic Systems Building Protocol. This is the case, for example, when an army unit is deployed on the battle field receives a new payload for one of their unmanned vehicles. Plugging in the payload will trigger a set of credential exchanges between the UAV and the new equipment, they then verify each other's identity and then the payload will upload its main key to the vehicle's hub. This procedure acts in the following way: the new device is connected with level of trust 0 and then uses the Anonymous Services to upload a certificate for the device public key and requests the certificate of the Hub for verification. Following a successfully credential validation on both sides, the device uploads a newly generated Main Key to the system encrypted with Hub's public key and signed with the Device's private key. This procedure however is applicable only for devices having non trivial computing power on board since it involves CPU intensive public key cryptography. The devices and the hub must be prepared to handle multiple levels of certificates. As presented in the hypothetical example above, the manufacturer of the vehicle may not know about the manufacturer of the payload and reversal. However, both manufacturers will be able to provide an army certificate for the key they used to sign the key of the device and vehicle respectively. This dual level of certificates allows full validation since both the vehicle and the payload knows the armies public key.

C. Packets and Messages

A sDOMO Packet is carried on a single UDP Datagram. Each packet contains a header followed by a variable payload section. The Packet ID specify the function of this packet. DID is set to 0 for packets that does not have yet assigned a Device ID like NewBabyBorn or Addoption Offer. Once a session is initiated the DID specify the assigned Device ID. Packet number is a unique value incremented at every sent packet, it is used to eliminate eventual duplicates on the network and to prevent reply attacks. Signature Type specify the HMAC algorithm used sign the packet. No signature=0, SHA1 HMAC=1, SHA256 HMAC=2. Encryption Type specify which encryption algorithm is used for the Packet: 0-No encryption, 1–XOR, 2-AES128, 3-AES256 are currently defined. The signature value is dependent on the signature type, for no signature the length is zero, i.e. the packet body starts immediately after the encryption type. For SHA1 based HMAC the signature is 20 bytes long and for SHA256 it is 32 bytes. Numeric values are packet in network order i.e. big endian format.

The Packet Body is also a collection of network order fields. Strings are packed as zero terminates ASCII string, the byte 0 being required at the end of the string. Arrays are prefixed with the size encoded as variable-length code then the specified number of items follows. sDOMO does not send floating point over the wire but recommends to the message designers to use fixed point format represented as integers; for example our thermostat demo sends the temperature in milli-Kelvin packed as an unsigned int 32 bit which is enough to encode any temperature reachable on Earth with precision more than enough for most practical applications.

Messages are sent as a sequence of one or more Message Carrier Packets. Each packet contains the start position of the current packet in the final message. Packets arriving in expected order are not ACK-ed until the final packet arrives and then the whole message is ACKed with a single ACK packet. However, any packet out of expected order triggers a negative ACK containing the expected sequence. The Hub will receive a whole message and ACK upon reception to the sender before being delivered to the intended recipient. It is important to notice that therefore receiving an ACK for a whole message does not mean that the receiver got it., That means a confirmation of delivery is required, the designer of an Interface must implement it at the message level.

IV. SECURITY AND PRIVACY CONSIDERATION

Home automation and robotic systems bring a whole new dimension to the problem of security. In a smart house, a potential attacker can take control and alter the physical aspects of the environment, potentially even inflicting bodily harm to the residents. In sDOMO we oppose the view of having each device individually connected to the Internet and we propose a House Centered, Self Sufficient domotic system. Putting the whole house network behind a firewall and controlling the access from outside via the Internet Gateway is the first step in security and privacy of the residents. However, as long as the desktop and mobile operating systems that needs to access the home network are still vulnerable, the threat of an intruder infiltrating behind the firewall is a likely possibility. To guard the domotic system against this probable scenario, a two tiered packet level and message level security measures had been designed as part of the sDOMO protocol.

A. Packet Level Security Considerations

After a new connection has been established and a Session Key has been downloaded from the Hub to the Device each sDOMO packet exchanged between a Device and the Hub is signed with a Hash based Message Authentication Code (HMAC). The HMAC is calculated with the formula:

$$HMAC(K,M) = H(K \mid H(K \mid M))$$

where H is the Hash function used, K is the Session Key and M is the message. The operator | specify byte stream concatenation. The Message subject to hashing consists on all the fields in the Packet Header (with the exception of the Signature itself) and all the fields on the Packet Body. In order to prevent reply attacks, when the highest PacketNumber of the incoming or outgoing packets is approaching the overflow limit, the Device shall re-send a Network Join packet that will

renew the SessionKey and reset the counters for PacketNumber to 0. By making sure that the combination (PacketNumber, Session Key) is unique and by not accepting Packet Numbers lower or equal than the previously received one, the system is protected against reply attacks.

To download a session key a Device is asked to generate a new 64 bit random number CR every time it sends a Network Join packet toward the Hub. The Hub will in turn generate its own 64 bit random number HR and encrypts the Session Key with a Download Key calculated as:

DldKey=SHA1(CR | Device Main Key | HR)

The HR is sent by the Hub toward the Device along with the encrypted key in the Join Accepted packet. The existence of both random numbers insure that the DldKey is unique every time a download happens and this prevents a reply attack in this stage in the connection process when a sequence number is not yet initialized. The fact that the DldKey is unique also allows usage of very small devices that have no encryption capabilities other than XOR. Since the encrypted SessionKey is a random number without any structure to reverse engineer having a size equal with the SHA1 hash, and since the Main Key is not known to the attacker the simple XOR with the unique DldKey is an approximation of one-time-pad encryption.

Streaming Interfaces as specified above are direct data connections between two Objects from different devices over the network, without the involvement of the Hub in communication process. The Hub however is involved in managing the security keys of the system. A procedure for allowing the Hub to manage and distribute the keys for streaming interface is being developed. While sDOMO allows for unsecured Streaming Interface this is a feature that needs to be thought about very carefully. For example if an attacker can spoof a thermometer sending data to a tablet, the worst that can happen is for the residents to get confused. However, if the same thermometer is also connected to the A/C control unit the results can have more serious implications. If a hacker sends to A/C unit a fake high temperature when the house is cold putting chilling in overdrive an eventual sleeping diabetic patient unable to properly sense the temperature may go into hypothermia. Therefore, for any control action secured communication via the Hub or encrypted Streaming Interface is highly recommended.

A unique security feature of sDOMO protocol, as far as we know, is the prevention of device kidnapping by the threat of disclosure of the attacker. If a dishonest neighbor or contractor is solicited to help with the installation of a new device, the installer will have access to the new Device Main Key in order to upload-it into the Hub. If the bad guy installs a Trojan Horse masquerading as a fake hub on the residents laptop there is nothing a cryptographic protocol can do to prevent the fake hub to take control of the device. However, as a deterrent for this kind of situations sDOMO specifications requires that the Network Join packet, containing the address of the Hub intended to be joined, to be sent by Multicast or Broadcast if the underlying network technology permits it. It also requires for any Hub to log all the Network Join events it receives and informs the House Intelligence Unit of anything it heard on the local network about something joining the hub that is not known to be part of this domain. That is, if a Trojan Horse attempts to hijack a device, this results in immediate disclosure of the Trojan and counter measures are going to be taken by the House Intelligence Unit.

B. Message Level Security Considerations

Attacks on a domotic system are not necessarily done by outlaws penetrating the network but also by the legitimate software or devices. We are living in a world where many companies, some of them very large, generate most of their revenue from collection and selling personal information to whomever is willing to pay for it. Some of these companies have already entered the field of Domotics and Robotics. While putting a small-print note in the user agreement may give to the vendor legal rights to snoop on other devices, our approach to privacy does not agree with this practice and sDOMO is designed to take some steps toward protecting user privacy by protocol design.

The Interface Definition Files specify for each message the Security Level. Security level 0 are messages which even if sent maliciously to a device have totally benign results while the messages at level 3 must be permitted only by the authorized system administrator. Similarly for outgoing messages, a level 0 means that any information in a message can be made public without any concern, while a level 2 or 3 respectively means that this piece of information must be kept secret and received only by authorized devices and administrative programs respectively. Similarly, any Device connected to the system has to have an associated level of trust from 0 to 3. The level of trust 0-Anonymous, is assigned to devices that connect to the Hub but which do not have a Main Key uploaded into the Hub. Beside Hub's Anonymous Services very few devices, if any, are expected to accept level 0 messages. When a device D1 sends a message to another device D2, or when D1 submits a request to subscribe to a notification emitted by D2 the message/request gets assigned a priority of:

MP=min(Device Trust Level, ACL(D1, D2))

where ACL(D1,D2) is returned from an Access Control List maintained in Hub's Database. If MP >= Security Level then the request is granted otherwise an error packet is sent back to the requester with an access denied error code. The ACL entries are assigned on a device to device basis, therefore a device wishing to snoop on the notifications sent by other devices will be denied unless the permission has been explicitly granted.

To facilitate the constructions of ACL's by regular home-owners without training in computer security, sDOMO requires that any device that needs to communicate as client with another must list the interfaces it needs to access in the Device Spec File. The <client> tag entry must also provide the level of access required for that device, if granting this right is mandatory for the functionality of the device and a clear text description of the reason why this access is required. The Console Application used to scan a new device will download the spec file and will initiate a user friendly dialog for granting the rights. If a particular interface is not listed on the spec file, no ACL entry will exist for that particular pair and therefore the access will not be possible for anything but Security Level 0. Of course a dishonest manufacturer can list in the Spec File all the possible devices it wants to snoop on, and here comes

the first line of defense: Shame. It does not require a high level security training to understand that a thermostat does not need access to all the indoor cameras and microphones to "regulate the temperature inside the house". It is hopefully that at least a small number of users to take their ire to blogosphere to make the manufacturer life harder. Of course, only reliance on shame alone is not good enough to protect the people privacy therefore sDOMO introduces a third XML file for Standardized Expert Advice. The Hub will maintain a list of web-sites and a database with PKI Certificates of experts publishing device reviews. These expert reviews will be presented as a signed XML files for each device that has been reviewed and will contain a set of proposed alternative ACL's each of them with an explanation of what privacy problems it solved and what functionality shortcomings it will generate. The Console Application used for installing a new device will automatically download the reviews and present the home owner with alternative configuration options based on the compromise privacy/functionality they are willing to make. The Expert Advice Files can be automatically scanned by the online stores and provide ratings of the devices before purchase. This direct feedback can have the effect of making the vendors more privacy conscious.

V. ADDITIONAL SERVICES

sDOMO protocol has been designed to be compact, fast and efficient to allow small footprint devices while conserving network bandwidth therefore in order to perform the complex functions required by a domotic system additional services and information are needed. To handle these requirements a set of configuration files and additional services are implemented beside the House Hub and home automation devices.

A. XML Files

The Device Spec file contains documenting features, capability specifications and default configuration values, links to interfaces exported by the device (as a server) and to interfaces imported by the device (acting as client). Documenting features allows for alternate language specifications for Device Name, short description and link to URLs for both user and developer documentation; icons for representing the device in management GUIs etc. Capability specifications contain the list of the configuration parameters of the Device, their valid ranges of values and default values for them. For example a sensor for the door or windows is powered by a coin cell battery will want to stay in sleep state for up to 24 hours to preserve battery life. This device needs to specify in the Spec File that it can go idle for up to 90000 seconds at a time otherwise the Hub may attempt to ping it after 10 seconds of inactivity and disconnect it after a failure to respond to 3 consecutive pings. Interface Definition Files are referred from the Device Spec file and describes all the Messages and additional data structures used to build a Message as understood by a given software Object used in the device implementation. In addition to this, the Interface File specifies how the Direct Messages defined above are combined to provide Remote Method Calls. The Interface Definition file it is used by an Interface Compiler that parses it and generates classes used to communicate with the device (Proxy) or implement the specified service functionality (Stubs). Manufacturer proposed rules for HUI are also listed here.

A third XML file used by our domotic system is the Standardized Expert Advice File. This file is used by the specialists analyzing the devices existent on the market in order to eliminate potential privacy threats to users. Beside alternative ACL and explanations for their use the expert advice files can contain alternative default settings for the device (overriding parts the Spec file) and conditions when they are a better use that the one provided by the manufacturer. Another entries that may be added to the expert files will be rating of the device in accordance to various criteria to allow buyers to make more informed decisions while shopping for a new device. Scripts for House Intelligence as well as Unit and Internet Gateway can also be offered as part of the expert advice files.

All the files above are XML files which are signed using Public Key Cryptography. We are experimenting currently with "GnuPG clearsign" signatures but on a commercial system XML signature as specified by the W3C XML-Sig proposal are expected to be available too. Every sDOMO Hub is expected to maintain a database of PKI certificates along with a list of servers for expert advice and methods for querying them for a given file.

B. House Intelligence Unit and Internet Gateway

The House Intelligence Unit (HIU) is a special device having close relationship with the Hub with which share access to the Hub Database. HIU is responsible for handling out of ordinary conditions that need special attention. This handling is achieved by a set of processing rules allowing HIU to monitor notifications emitted by various devices, and when certain conditions are met emit alerts or handle the condition by either using plug-ins or scripting. To aid the automated building of the HIU rules for a device, the manufacturer can add into the interface definition file potential alerts which upon a device start-up are loaded by the HIU. Besides rules from interface files, HIU can have its own set of rules added by the

```
<intelligence>
    <meta_alert notification="CycleCompleted" criticality="Low" domains="House" type="text">
        <condition>(msg.Counter >= 8) and (msg.Counter % 4 == 0)</condition>
        <text lang="en">Washing machine needs your attention</text>
        <text lang="ro">Masina de spalat are nevoie de atentia dumneavoastra</text>
        <append_msg_text msg_text="Text" />
    </meta_alert>
</intelligence>
```

Fig. 3. Intelligence tag for washing machine example

console from a system administrator or from the expert advice files for this device.

In the simple example from the figure above, HIU subscribes for the CycleCompleted notification emitter by the washing machine. Based on the condition tag, when 4 minutes passed and the notification is still being braodcasted HIU will start emitting a text alert with the first line of text: "Washing machine needs your attention" and having on the next line the content of the field Text from the notification sent by the washing machine. Since HIU alerts are most likely displayed by all GUI based devices into the house, the distracted housekeeper will be notified about the washer need for attention while watching her favorite show on the smart-TV.

By specifying various criticality levels and domains of actions, HIU can alert a nurse about the fall of a resident as detected by a computer vision enabled surveillance camera or send to the home-owner vacationing out of town a text message in the same time with calling the fire-fighters with a prerecorded message if the smoke detector and heat sensors

simultaneously detects activity in a room. Scripts in HIU can also monitor the humidity level of the soil, query the weather channel web site for the forecast and decide if to turn on the irrigation or not in the garden, etc.

HIU is also the first line of defense against device hacking, being the one alerting the users and the company installing the system about the attempt of device hijacking as detected by sDOMO device hijacking prevention schema.

The Internet Gateway is the only point of access from the outside world into the domotic system. The house automation network contain sensitive information and can harm the residents if misused with intention or by accident. However, the home owner should be able to watch his surveillance cameras when a motion alert text message is received or a nurse should be able to take control of a robot from an independent living assisted house when an emergency condition has been detected. To allow for this contradictory requirements the Internet Gateway will be able to tunnel sDOMO messages toward devices located outside the house network. The communication is taking place over an encrypted network connection and the tunneling connection it is opened only for limited amounts of time subject to periodic re-authorization between the remote device and the gateway. The original authorization for opening the connection must be done via a separate communication channel like HTTPS or SSH and a Tunneling Session Key it is issued for the other end of the tunnel (the Remote Device Service). The RDS in turn will establish an encrypted communication with the Gateway encrypting the channel with the Tunneling Session Key and authenticating with the gateway with a separate key pre-assigned to that device. Once the tunnel is opened the Device software located in the mobile device will be able to connect to the Hub like any regular sDOMO device from the local network.

VI. CONCLUSION AND FUTURE WORK

We successfully demonstrated the implementation of a native speaking sDOMO thermometer based on an Arduino Uno (2 KB SRAM and 32 KB Program Flash) a WS5100 "Ethernet Shield", TMP-102 I2C temperature sensor and LEDs as placeholders for controls. This implementation proved that the sDOMO it is scalable enough to accommodate small devices as full nodes on the network while sending packets authenticated with SHA1 HMAC. The devices emits a notification either as a reply to a direct message from the GUI either once every second if no control message arrived. The notification message contains both the temperature in milli-Kelvins and the state of the three controls pins, connected in the demo to first 3 LED's. The control message from the GUI contained a new state of the control pins which turns on or off the LED's connected to the pins. The demo source code and schematic for the Arduino thermostat is available on the companion website.

The second demo from the companion website implements an elementary file transfer server and client pair which allowed us to perform a set of performance measurements by comparing the download speed achieved by our demo against the FTP software that came with Mageia Linux (pure-ftp for server and lftp as client).

TABLE I. FILE TRANSFER PERFORMANCE

File Size (bytes)	Transfer Speed MB/s	
	sDOMO Demo	*FTP Software*
11461	0.91	N / A
3785652	1.48	1.99
77594624	1.48	2.10

The measurements shows that our non-optimized "proof of concept" demo program performed decently while compared with the mature, highly optimized FTP software for Linux despite the fact that the FTP it is a direct transfer between the client and server while in our demo the messages carrying the files were passed between the server and client using the House Hub as intermediary. If we also consider the fact that FTP it is an unsecured connection, vulnerable to a "Man In the Middle" (MiM) attack while all sDOMO packets were signed with SHA1 HMAC and verified at both ends making such an attack impossible, we see that our claims about the efficiency of sDOMO protocol are validated by this measurements.

Implementation of Console Application, HIU and Internet Gateway are still in work as of this moment.

This paper presented an introduction of the sDOMO protocol and its associated system architecture. The scope of this paper was to provide an insight on our ideas and it is not an exhaustive protocol specifications document. To fully define the protocol, offer downloads, tutorials and demos a companion site of this paper can be found at:

http://www.mezonix.com/sDOMO.html

REFERENCES

[1] Marcel-Titus Marginean and Chao Lu, "A Distributed Processing Architecture for Vision Based Domestic Robot Navigation", 2013 International Conference on Computers, Communications and Systems, Korea.

[2] Theis Solberg Hjort, Rune Torbensen, "Trusted Domain: A security platform for home automation", Elsevier, Computers & Security 31 (2012) 940-955.

[3] Thinagaran Perumal, A. R. Ramil, Chui Yew Leong, "Interoperability Framwork for Smart Home Systems", 0098 3063/11 2011 IEEE.

[4] Yung-Wei Kao, Shyan-Ming Yuan, "User-configurable semantic home automation", Computer Standards & Interfaces 34 (2012) 171-188.

[5] Pedro Sanchez et al., "A framework for developing home automation systems: From requirements to code", The Journal of Systems and Software 84 (2011) 1008-1021.

[6] Poul Ejnar Rovsing et al., "A Reality Check on Home Automation Technologies", Journal of Green Engineering 303-327 2011.

[7] Vittorio Miori et al., "An Open Standard Solution for Domotic Interoperability", 0098 3063/06 2006 IEEE.

[8] Juing-Huei Su et al., "The Design and Implementation of a Low-cost Programmable Home Automation Module", 0098 3063/06 2006 IEEE

[9] Ali Ziy Alkar et al., "IP Based Home Automation Systems", 0098 3036/10 2018 IEEE.

[10] Marcel-Titus Marginean and Chao Lu, "A Multi-paradigm Object Tracker for Robot Navigation Assisted by External Computer Vision", ACM RACS 2014.