### Title

Security and Privacy for Emerging Smart Community Infrastructures




### Authors

Copos, Bogdan
Levitt, Karl
Rowe, Jeff
et al.



### Publication Date

2016-04-01

Peer reviewed

# Security and Privacy for Emerging Smart Community Infrastructures

Bogdan Copos[1], Karl Levitt[1], Jeff Rowe[1], Parisa Kianmajd[1], Chen-Nee Chuah[1] and George Kesidis[2]
[1]*University of California, Davis, Davis, CA, U.S.A.*
[2]*Penn State University, State College, PA, U.S.A.*
{*bcopos,knlevitt,jbrowe,pkianmajd,chuah*}*@ucdavis.edu, gik2@psu.edu*

Keywords: Cyber-security, Privacy, Internet-of-Things, Microgrids, Software Defined Networks (SDN), Cloudlet

Abstract: Smart communities of the future have features that make them susceptible to novel forms of cyber-attack and a potential loss of privacy for the citizens they serve. We view these communities as metro level wide area control systems with sensors and actuators located in residences, the workplace, in mobile vehicles and even worn on the body. In addition, this distributed system may not be subject to centralized control. It needs to be responsive to the individual needs of citizen owners yet still maintain the ability to coordinate actions across a neighborhood, or larger metropolitan area. The question we wish to address is, as frameworks emerge to handle these unique challenges, how can we provide security and privacy for such and open and decentralized environment? We suggest ways to add security and privacy to low level IoT devices, to a cloudlet based application platform, to a wide area SDN for coordination, and to negotiation protocols for citizen coordination.

## 1 INTRODUCTION

By connecting rapidly emerging cyber-enabled sensors and actuators and providing communications protocols, a wide variety of novel beneficial applications become available. Communicating position and speed between vehicles and with the roadside infrastructure can enable optimal highway traffic management (A. Thiagarajan and Eriksson, 2009; Hounsell et al., 1998; James, 1995; Amoozadeh et al., 2015; A. Chen and Zhang, 2006; Khorashadi et al., 2011; B. Liu and Zhang, 2010) applications, including smart intersections and efficient vehicle platooning. Wearable health monitoring devices that communicate physical activity, heart rates and other biomedical data to health care professionals provides for new health maintenance and independent living lifestyles. Coordination of residential solar arrays, battery storage, and plug-in electric vehicle chargers would allow novel distributed optimal microgrid control schemes (Alizadeh et al., 2013b; Alizadeh et al., 2014a; Alizadeh et al., 2014b; Alizadeh et al., 2013a; Lu et al., 2013) for neighborhoods. It is critically important, however, that the introduction of these new technologies doesn't introduce new cyber-security vulnerabilities and threats to citizens' privacy.

As new architectures are designed and developed, provisions for robustness and resiliency in the presence of both natural faults and malicious attack must be included. Emerging IoT routing protocols include self-healing and tolerating intermittent loss. Similar treatment for cyber-security and privacy mechanisms should be built in from the outset. This includes separation of privilege, network and device access control, and runtime monitoring among other things. Towards a general smart community secure framework, we define a smart community metro-scale network as the infrastructure that enables coordination and control over a wide variety of cyber-physical systems. This framework includes,

- Internet of Things (IoT) devices and communications protocols - low-powered, resource poor physical sensor and actuator devices with limited communications capabilities.
- Mobile Cloudlet Infrastructure - an intermediate software application infrastructure that is located very close to the IoT or mobile devices, can handle resource intensive computations, can migrate through the network with mobile devices and has a high-quality connection to standard fixed cloud services.
- Smart Community SDN - an SDN backbone running over the wired Internet that connects cloudlets and provides tunneling of IoT protocols for mobile IoT devices.

Figure 1 gives a notional overview of this architecture. Individual buildings or sites maintain an in-

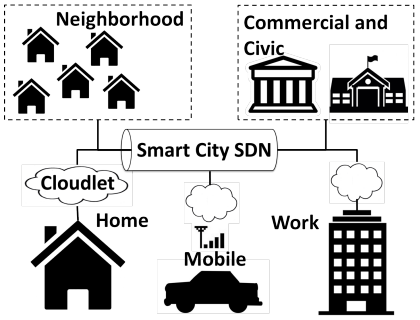


Figure 1: A notional smart community network architecture.

ternal IoT device network. IoT devices are networked to communicate with the site "root" gateway using the RPL routing protocol designed specifically for such networks that is rapidly becoming a standard for IoT. The Internet ingress location is provisioned to host a cloudlet that manages compute intensive applications to provide high level control of a site's devices. These cloudlets can migrate from ingress to ingress point for mobile devices as they travel. A software-define network (SDN) is used over the wired Internet backbone to enable mobile IoT devices to operate seamlessly with the fixed site devices and to provide access control and isolation between sites and applications. Cooperating sites use the SDN to implement metro scale control applications that use IoT devices of all users as sensors and actuators. Present consensus is that SDN is used to simplify network configuration including virtualized network functions (VNFs). VNFs could include intrusion detection systems and firewalls deployed at the site's Internet gateway, WiFI router or at a nearby cloudlet (the cloudlet generally in support of mobile/cellular communications). For the remainder of this paper, we discuss how each component of this framework would be used and how security and privacy concerns can be addressed.

# 2 SECURITY OF THE LOCALIZED SITE NETWORK

The first link in a smart community control network is the connection to low-power, low-resource IoT devices that serve as sensors and actuators at a particular site, like a residence, a commercial building or public property. We assume that each of these sites will have some gateway to a high-capacity, high-availability network backbone, either wired or wireless. The question is, how can we avoid introducing a new potential attack vector of the localized IoT site network with this IoT device routing protocol. The routing protocol must support low power IoT devices and efficiently handle lossy ad-hoc networks constructed by such devices. The protocol must also allow for flexibility on the criteria used for optimizing routing. For example, in some scenarios, routing should be optimized based on the number of hops while in other cases, the reliability of the transmissions may be more important. With the emergence of IoT devices, the Internet Engineering Task Force (IETF) recognized the need for a standardized IPv6 routing solution for IoT networks. As a result, a working group was formed and assigned with this task. After careful analysis and several experiments, the group produced RPL. The RPL protocol has since been adopted and serves as the standard routing protocol for low power lossy networks (LLNs) (Vasseur et al., 2011). Its role as the standard routing protocol in LLNs and useful features make it an ideal candidate for our framework.

## 2.1 RPL Routing Protocol Overview

RPL is a routing protocol for low-power lossy networks in which nodes are connected through multi-hop paths to a root device, creating a Destination Oriented Directed Acyclic Graph (DODAG). The root device is typically a hub or border router that has some limited responsibility for localized device coordination and that serves as a gateway to a more reliable high-speed network. Every node in a DODAG has a rank which is computed using an Objective Function (OF). The rank indicates the position of a node with respect to the DODAG root. Consequently, ranks strictly increase with the number of hops from the root. The Objective Function defines routing constrains and other properties taken in consideration during topology construction.

In the initial phase of topology construction, nodes listen for DODAG Information Objects (DIO) messages which includes RPL instance ID, IPv6 address of the root, etc. Nodes process these messages and, depending on their preference, select a DODAG to join. The first step of joining a DODAG is selecting a parent. During parent selection, the node scans for nodes in its range and selects the node with the best (i.e. lowest) rank as the parent. After parent selection, the node computes its own rank using the OF and sends Destination Advertisement Object (DAO) messages. DAO messages are used to propagate destination information upwards, toward the root.

Since every node maintains information about their parent, upward routing is straight-forward.

However, to enable downward routing, RPL has two modes of operations: nonstoring mode and storing mode. In nonstoring mode, each packet contains the route the packet is expected to follow through the network. The route is computed by the root, which is required to maintain information about each node in the network. On the other hand, when storing mode is used, RPL uses stateful in-network routing tables. In other words, every node in the network keeps routing tables to differentiate between the packets heading towards the root and the packets heading away from the root. Through the use of upward routing, downward traffic, or a combination of the two, RPL supports various kinds of traffic including Multipoint-to-Point (MP2P), Point-to-Multipoint (P2MP), and Point-to-Point (P2P).

RPL also has a number of other useful features, including self-healing capabilities. There are two repair mechanisms, global and local, which are applied if a link or node failure is detected. If a link or node failure is detected, the local repair mechanism will attempt to identify a new parent or path. If multiple local failures occur, a global repair will be performed where the entire DODAG is rebuilt. Furthermore, RPL uses a trickle timer to efficiently handle inconsistencies such as loops, joining of new nodes, rank changes. If the network is stable, the trickle timer interval is large. An inconsistency will cause the timer to be reset and DIO messages to be transmitted.

## 2.2 RPL Security

Despite having loop detection and self-healing capabilities, the RPL protocol is not equipped with any protection mechanisms against traditional routing attacks. In (Wallgren et al., 2013) the authors show that RPL is vulnerable to a variety of traditional routing attacks, including sinkhole, selective forwarding, blackhole, wormhole, HELLO flood, and clone ID attacks. All these arise because of the ability of nodes to advertise false ranks. For example, an attacker can falsely advertise a beneficial rank thus making the neighboring nodes route traffic through it. Once traffic is routed through the attacker, the attacker can eavesdrop or execute Denial-of-Service by dropping all packets it receives. A HELLO flood attack can be executed using DIO messages, which are used to advertise information about DODAGs to new nodes. In such an attack, a malicious node with strong signal power sends DIO messages to various nodes on the network causing those nodes to route traffic through it. Since most of the benign nodes do not have an equally strong signal power, their messages will not be successfully transmitted. Unfortunately, the built in self-healing mechanism, designed to deal with natural faults, is ineffective against the malicious attacks. While in some cases, given enough time, the self-healing process helps mediate the attacks, it is never able to fully eliminate the effects of the attacks.

We are exploring two different approaches to lightweight monitoring of IoT ad-hoc routing protocols. In the first approach, we use ideas borrowed from our previous work in monitoring wired networks (Cheung and Levitt, 1997). This approach uses conservation-of-flow to identify and isolate misbehaving routing nodes. Each routing node maintains a sum of the sizes of all network packets that it sends and receives on each link. Periodically, this value is shared with other neighboring routing nodes in the network. After receipt of flow summaries, each device can compute the net sum of a all traffic entering and exiting a subset of the ad-hoc network graph, and if the net value is sufficiently far from zero, it can be assumed that a node in the subset is either dropping packets or injecting false traffic. This approach relies upon having multiple paths to a specific device for data sharing and for providing an alternate route if isolation of the subset is required. Since RPL generates a tree routing structure directed at the root node, we use lightweight cryptographic authentication to sign shared flows, preventing a malicious device in the path from supplying false information.

Since IoT devices are supplied by third-parties and have computing power limited to a specific application, implementing such a ubiquitous verification scheme would not be practical. We also consider including an IoT device whose sole purpose is to provide runtime monitoring, without necessarily serving as a physical sensor or actuator. These low-cost, low-power IoT security devices would need to be located on enough routing paths to be able to isolate subsets of the RPL network. We are investigating modifications to the RPL routing OF that includes not only optimal connectivity, but also path security metrics. Routing node rank would be higher for paths that include one of these IoT security monitors. A security monitor device that sees rank advertisements from other security devices could dynamically alter its rank to be lower to ensure that not all monitors cluster in one isolated path. In this manner, routing integrity can be maintained against both natural and malicious faults even in networks of off-the-shelf devices programmed without any specific routing security in mind.

# 3 SECURITY AND PRIVACY WITH A MOBILE CLOUDLET GATEWAY

Current commercial IoT devices, such as the Nest family of products, rely heavily upon a centralized cloud service for configuration and control. The current generation of devices are configured by the user only with credentials necessary for accessing a site's WiFi gateway. Once a network connection to the cloud controller is established, no other on-site computing resources are involved. This architecture leads to several security and privacy concerns. First, attackers who compromise the central cloud controller will own all devices on all sites and have access to users' information. The IoT devices send a citizen's personal and potentially sensitive information to a third-party which may have different interests. Network disruptions can result in significant loss of functionality if a device is unable to acquire important control information in a timely manner. Other IoT devices, such as the SmartThings family, use a dedicated central controller device. The controller is connected to the site's Internet gateway and serves as the communications hub for all other SmartThings sensors and actuators. Such an architecture allows the device owner to have more control over personal privacy and is more robust to external disruptions to the Internet. However this presents its own set of drawbacks. Mobile IoT devices, such as wearable health monitors, personal key-fob trackers or vehicle based devices will often times be out of range of the central hub. Some applications might tax the power of a dedicated communications hub which isn't designed for general purpose computing, such as voice or image recognition.

We investigate a mobile cloudlet architecture that can provide resource intensive compute services to the IoT devices under the control of a single site, regardless of where they are located geographically. Cloudlets are an emerging platform for handling mobile high-performance applications. Currently, common cellphone applications, such as map navigation, voice and image recognition and language translation are run by cloud services at locations widely separated from the device. When a user is moving, this can lead to service degradation as the path to the cloud is constantly updated. Cloudlets have been proposed that run on provisioned hardware located at first-hop Internet ingress points.

Cloudlets are virtual software containers that migrate from location to location as the user moves about so that high performance can be maintained for compute intensive services. These containers can also be used to supply security and privacy services to both fixed and mobile devices of a site in a smart community. Since the cloudlet isn't controlled by a single 3rd party, the exposure of potentially sensitive user information would require gaining access to that user's site cloudlet. The information of all smart community citizens is distributed across multiple sites under the control of the citizens themselves, providing a greater measure of isolation and separation of privilege.

The RPL protocol monitoring previously discussed protects the routing infrastructure but doesn't address attacks on the control application itself. Application level runtime monitoring requires significant computing resources to analyze message content and to track the state of devices. We investigate cloudlet based application monitoring based upon physical specifications we call "rationality checks", borrowed from the automotive industry. Rationality checks track the state of physical devices based upon application protocol messages sent from sensors and control messages sent to actuators.

Two types of consistency checks are possible. First, checks on the consistency of protocol state can detect malicious message injection that attempts to subvert IoT devices. For example, suppose an application is designed so that a temperature sensor request is sent every time the motion detector is triggered so that the A/C can be turned on only when the occupant is at home. An attacker that has compromised a site device might be able to forge a false temperature report to activate the A/C even if the occupant is absent. The rationality check in this case would see that a temperature response without a corresponding motion detector request violates the state specification of the application protocol. Another type of rationality check is a constraint based upon the physical properties of the system. A message from the thermostat showing a 20 degree temperature rise in the course of several minutes, for example, would be an obvious violation of the physical properties driving the sensor; it would be impossible for the building heater to increase the temperature by that amount in a short time period. A cloudlet based application gateway provides both privacy of IoT generated information as well as a platform for performing compute intensive cyber-security analysis.

# 4 SECURE SDN FOR SMART COMMUNITIES

Smart community applications will use communications between sites and mobile users to coordinate specific activities of the community's citizens. This

coordination can be disrupted by attackers targeting the smart community backbone network connection, through denial-of-service, injection of malicious packets or eavesdropping on sensitive communications. We envision using an SDN that can provide enhanced security services and countermeasures when attacks are recognized.

An SDN can add to the security of the network, but along with this flexibility comes added risk. Authorization and authentication in the SDN switch are more complicated. For example, commands reconfiguring the switch must come from a trusted, authorized source, as must updates. SDN switches can also provide several security services to protect their networks. Generated statistics are more accurate than those supplied by ISPs, which can provide better data to an intrusion detection system or rationality checker. OpenFlow Random Host Mutation dynamically allocates a random virtual IP address mapped to the real IP address, hiding the real IP address from external sites which can protect the privacy of citizens. The controller can enforce dynamic access control policies based on flow-level information to prevent emerging denial-of-service attacks in their earliest stages. Finally, applications can act as edge-based network authentication gateways to ensure that forged packets claiming to originate at false smart community sites are recognized and blocked.

We are also investigating the use of the SDN to provide a seamless IoT environment for mobile devices. Low-power, low resource IoT devices such as motion trackers and heart rate monitors may not be able to recognize the context of their environment to alter their behavior. For example, a health monitor that travels with a user from home to the workplace might still need to communicate with devices that remain in the home. The mobile cloudlet traveling with the user provides the high-level application controller but not one that includes other stationary devices in other locations. We investigate SDN protocols for tunneling low level RPL type routing messages through the backbone network so that resource limited IoT devices maintain their application context regardless of physical location within the site.

# 5 PRIVATE DECENTRALIZED SMART COMMUNITY COORDINATION

A final component of secure and private smart communities is the coordination mechanism among community citizens. In order to provide enhanced services and optimize the usage of shared resources, citizens must have some method to reach agreement on a collection of actions. This might be explicitly constrained by the application, such as fairness in wait times at a smart intersection, which doesn't require any citizen interaction. Other applications require a period of negotiation before an agreement can be met. We choose a virtual smart electric microgrid as our application. It is well known that solar power arrays suffer from the problem of inelastic demand. The power is generated based upon solar radiance, which doesn't match immediate demand. This problem will only become worse with the increase in plug-in hybrid electric vehicles which begin their charging cycle when the sun has started to set in the evening. Residential power storage in the form of batteries are already becoming available, which can alleviate some of the problem.

To optimally use these devices we consider a smart community power sharing application, which serves as a virtual microgrid. Citizens in a neighborhood have a variety of renewable power devices and a variety of usage load profiles. By sharing generation and storage resources and agreeing upon a usage schedule, the collection of devices can be optimized to the advantage of all participants. The question is, how can citizens coordinate this activity without revealing details of their power use to others, which is a major privacy concern today. Analyzing a site's energy usage information can leak private their household appliance use data. By analyzing your energy usage one can find out about kind of appliances used, time of use, as well as, when you are or are not at home or asleep. We are investigating a novel private coordination mechanism that is applicable to a wide variety of smart community applications based upon cryptographic blockchains similar to those used in digital currencies such as BitCoin.

## 5.1 Decentralized Coordination using Blockchain

The current literature has shown that applications of blockchains can go far beyond the financial domain. From distributed storage for private information, to distributed online marketing, blockchain can be used to provide distributed audit records (Zyskind et al., 2015). Blockchain distributed agreement is a good choice for smart community negotiation and coordination protocols since it can provide the means to build and maintain a distributed storage of privacy-sensitive energy usage information without a single owner. Also, blockchain provides us with an append-only immutable records which is essential for our sys-

tem and ensures the users cannot insert fake points to their energy usage points history.

The general idea is that similarly to the current power grid, generation and load scheduling is performed in advance. Typically this is done with a week-ahead market that matches bids to offers. The schedule is updated daily as conditions change, with a final generation and usage schedule generated 5 minutes ahead of runtime. This is currently performed by a central regional power authority in today's grid. The same type of advanced scheduling can be performed in our neighborhood microgrid using distributed bidding with a blockchain. The privacy of citizens would be maintained using random nonce identifiers generated by sites. Bids would be added to the blockchain and forwarded to all cooperating sites. The blockchain mechanism guarantees that bids cannot be altered or forged just like in the Bitcoin digital currency. All sites share the same bid/offer blockchain and can compute an optimal schedule which they also share. Agreements to the schedule are appended until all participants reach consensus. Throughout this process, individual site usage need not be revealed. Additionally, bids and agreements cannot be modified. Details of this scheme are given below.

## 5.2 Cryptographic Building Blocks

### 5.2.1 Commitment Scheme

A commitment scheme is the cryptographic relative of an envelope and aims at temporarily hiding a value, while ensuring that it cannot be changed later (like a sealed bid at an auction). A commitment scheme consists of two steps: forming the commitment and verifying the commitment. We use the notation $c = Com(m;r)$ to mean that $c$ is a commitment to the message $m$ using some randomness $r$ (called *decommitment key*). The commitment $c$ is like a sealed envelope with two properties: *hiding*, and *binding*, which state that the committer is the only one who can unseal the envelope, and the receiver will not be able to learn anything about its contents before the committer reveals it. To reveal the commitment, the committer has to reveal the random de-commitment key $r$ to the receiver.

We use "Pedersen" commitment scheme (Pedersen, 1991) in our system, that has *homomorphic* property such that if $c_1$ is a commitment to $m_1$ with randomness $r_1$, and $c_2$ is a commitment to $m_2$ with randomness $r_2$, then $c_1.c_2$ will be a commitment to $m_1 + m_2$ with randomness $r_1 + r_2$. Below comes the description of Pedersen commitment scheme (Pedersen, 1991):

- **Setup:** receiver chooses:
  - large primes $p$ and $q$ such that $p = 2^w q + 1$ for $w \geqslant 1$.
  - Random generators $g, h$ such that $\mathbb{G} = \langle g \rangle = \langle h \rangle$ and $\mathbb{G}$ is a subgroup of $\mathbb{Z}_q^*$.
    - Values $p, q, g, h$ are public, $r$ is secret
- **Commit:** to commit to some $x \in \mathbb{Z}_q^*$, the committer chooses random $r \in \mathbb{Z}_q^*$ and sends $cl = g^x h^r \bmod p$ to the receiver
- **Reveal:** to open the commitment, sender reveals $x$ and $r$, receiver verifies that $cl = g^x h^r \bmod p$

### 5.2.2 Discrete Logarithm Problem

Pedersen commitment scheme is based on the difficulty of finding an answer to discrete logarithm problem which is described below: Given $g^x \bmod p$ where $p$ is a large prime, it is "difficult" to learn $x$ (i.e. there is no known polynomial-time algorithm to learn $x$). $g$ is a generator of a multiplicative group $\mathbb{Z}_q^*$.

## 5.3 Our Framework

**Bidding Phase**

Each household $j$ makes an estimate $x_j$ for the amount of energy they will need for the next energy sharing time period and sends the commitment to that $b_j = com(x_j, r_j)$ together with the decommitment key $r_j$ to the blockchain.

**Allocation Phase**

The system decides if requests for energy can be fulfilled. First it will ensure that the amount of energy requested by each user does not go beyond a specific limit. We rely on the range proofs of Boudot (Boudot, 2000) and Camenisch (Camenisch et al., 2008) for this step.

**Recording Phase**

Let $e$ be a symmetric encryption. Periodically, the tamper proof meter installed for each household $u_j$, sends values of $p_i = e(x_i || timestamp)$ to the blockchain, where $x_i$ is the current energy usage at time $timestamp$. The meter also sends the commitment to that value $cu_j = (comm(p_i, r_i))$ off to the blockchain.

**Verification Phase**

Let $x_{total}$ be the total energy that was consumed at the end of energy sharing time period by the whole neighborhood. The coordinating algorithm first checks this amount does not exceed the sum of the requested energy by each of the households. Using homomorphic characteristic of our commitment scheme, we check if the following equation is held.

$com(x_{total}, \sum_{i=1}^{n} r_i) = \prod_{i=1}^{n} c_i$

Where $x_1, x_2, ..., x_n$ are the bids by the households and $r_1, r_2, ...$ are the decommitment keys they each provided.

**Evidence Gathering Phase**

If the result of verification phase shows that the total energy used in the neighborhood is more than the sum of the requested energy by each household, we can conclude that one (or more) household have used more energy than what they had asked for. To find those who disobeyed the protocol, we should check if the total energy used by each household matches with the amount that they had asked for in the *Bidding* phase.

$\forall user\ u_i : (b_j = \sum cu_j) \rightarrow 0, 1$

## 5.4 Cryptographic Construction

Our scheme is secure under the Strong RSA and Discrete Logarithm assumptions, and the existence of a zero-knowledge proof system.

We now describe the algorithms:

- `Setup`$(1^{\lambda}) \rightarrow params$. Generate $p, q$ large primes such that $p = 2^w q + 1$ for $w \leq 1$. Select random generators $g, h$ such that $\mathbb{G} = \langle g \rangle = \langle h \rangle$ and $\mathbb{G}$ is a subgroup of $\mathbb{Z}_q^*$. Output *params* = $(p, q, g, h)$.
- `Bid`(*params*, $x$)) $\rightarrow (c_x, r_{xi})$. Select $r_x \leftarrow \mathbb{Z}_q^*$ and compute $c_x \leftarrow g^x h^r \bmod p$ and output $(c_x, r_x)$.
- `Verify`$(params, x_{total}, r_{total}, C_{total}) \rightarrow 0, 1$. Given a total energy usage $x_{total}$ in the entire neighborhood, if $x_{total} = g^{c_{total}} h^{r_{total}}$ output 1, otherwise output 0.

Our protocol assumes a trusted setup process for generating the parameters. We stress that the accumulator trapdoor $(p, q)$ is not used subsequent to the *Setup* procedure, and can therefore be destroyed immediately after the parameters are generated. Alternatively, Sander describes a technique for generating accumulator parameters without a trapdoor (Sander, 1999).

# 6 CONCLUSIONS

When designing architectures and frameworks for next generation smart communities and associated metropolitan-scale networks and applications, it is essential to build in security and privacy mechanism at the outset. This includes security and privacy at the IoT device level, at the controller application level, at the network level and built in to the coordination protocols themselves. We have presented our ongoing work in designed-in security mechanisms for the RPL protocol that connects low-power, low-resource IoT devices. We also investigate novel cloudlet based architectures that can provide a measure of privacy to citizens whose devices are subject to the application control, and that can serve as the platform for resource intensive security monitoring and analysis. A smart community SDN can provide isolation, protection against denial-of-service and provide application context to low-resource mobile IoT devices without extensive reconfiguration. We finally present a private and secure method for distributed coordination and agreement based upon a shared blockchain data structure.

# ACKNOWLEDGEMENTS

We wish to acknowledge the generous support of the UC Davis RISE initiative and the NSF SaTC program which funded portions of this work.

# REFERENCES


A. Chen, B. Khorashadi, C.-N. C. D. G. and Zhang, M. (2006). Smoothing vehicular traffic flow using vehicular-based ad hoc networking amp; computing grid (vgrid). In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 349 –354.

A. Thiagarajan, L. Ravindranath, K. L. S. M. H. B. S. T. and Eriksson, J. (2009). Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 85–98, New York, NY, USA. ACM.

Alizadeh, M., Scaglione, A., Applebaum, A., Kesidis, G., and Levitt, K. (2014a). Reduced-order load models for large populations of flexible appliances. *IEEE Transactions on Power Systems*, PP(99):1–17.

Alizadeh, M., Scaglione, A., and Kesidis, G. (Nov. 2013a). Clustering consumption in queues: A scalable model for electric vehicle scheduling. In *Proc. IEEE Asilomar Conference, invited session on Signal Processing for the Smart Grid*.

Alizadeh, M., Scaglione, A., and Kesidis, G. (Oct. 2013b). Scalable model predictive control of demand for ancillary services. In *Proc. IEEE SmartGridsComm*, Vancouver, BC.

Alizadeh, M., Xiao, Y., Scaglione, A., and van der Schaar, M. (2014b). Dynamic incentive design for participation in direct load scheduling programs. *IEEE Journal of Selected Topics in Signal Processing*, PP(99):1–1.

Amoozadeh, M., Raghuramu, A., Chuah, C.-n., Ghosal, D., Zhang, H. M., Rowe, J., and Levitt, K. (2015). Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. *Communications Magazine, IEEE*, 53(6):126–132.

B. Liu, B. Khorashadi, D. G. C. C. and Zhang, M. (2010). Assessing the vanet's local information storage capability under different traffic mobility. In *INFOCOM*, pages 116–120.

Boudot, F. (2000). Efficient proofs that a committed number lies in an interval. In *Advances in CryptologyEUROCRYPT 2000*, pages 431–444. Springer.

Camenisch, J., Chaabouni, R., et al. (2008). Efficient protocols for set membership and range proofs. In *Advances in Cryptology-ASIACRYPT 2008*, pages 234–252. Springer.

Cheung, S. and Levitt, K. N. (1997). Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection. In *Proceedings of the 1997 New Security Paradigms Workshop (NSPW)*.

Hounsell, N., Landles, J., Bretherton, R. D., and Gardener, K. (1998). Intelligent systems for priority at traffic signals in london: the INCOME project. Number 454. IEEE.

James, R. D. (1995). Automated highway system. EP Patent 0,683,911.

Khorashadi, B., Liu, F., Ghosal, D., Zhang, M., and Chuah, C. (2011). Distributed automated incident detection with vgrid. *Wireless Communications, IEEE*, 18(1):64 –73.

Lu, H., Pang, G., and Kesidis, G. (Oct. 2013). Automated scheduling of deferrable PEV/PHEV load by power-profile unevenness. In *Proc. IEEE SmartGridsComm*, Vancouver, BC.

Pedersen, T. P. (1991). Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in CryptologyCRYPTO91*, pages 129–140. Springer.

Sander, T. (1999). Efficient accumulators without trapdoor extended abstract. In *Information and Communication Security*, pages 252–262. Springer.

Vasseur, J., Agarwal, N., Hui, J., Shelby, Z., Bertrand, P., and Chauvenet, C. (2011). Rpl: The ip routing protocol designed for low power and lossy networks. *Internet Protocol for Smart Objects (IPSO) Alliance*, 36.

Wallgren, L., Raza, S., and Voigt, T. (2013). Routing attacks and countermeasures in the rpl-based internet of things. *International Journal of Distributed Sensor Networks*, 2013.

Zyskind, G., Nathan, O., and Pentland, A. (2015). Decentralizing privacy: Using blockchain to protect personal data. In *Security and Privacy Workshops (SPW), 2015 IEEE*, pages 180–184. IEEE.