Secure Initial Configuration of Resource Constrained Wireless Smart Objects

Colin P. O'Flynn
IEEE Member. coflynn@ieee.org

***Abstract*—A lack of universal initial configuration (or *bootstrapping*) of low-cost resource constrained wireless sensor networks (WSNs) has greatly limited interconnectivity. Traditional methods such as a central authority or a user-entered passphrase are unacceptable on very resource-constrained nodes. Many methods have been proposed; most of them only suitable in specific situations. An analysis of common problems in WSNs will lead to a set of requirements for a bootstrapping system. From there existing solutions will be compared to the requirements; the results of which are a selection of four main methods of bootstrapping. The methods are then expanded on to describe considerations for deploying them in a secure manner.**

***Index Terms*—wireless, configuration, commissioning, bootstrapping**

I. Introduction

LOW cost wireless sensor networks are an expanding area of interest. The amount of effort shown in research papers, silicon, alliances, and standard organizations shows this dedication. A fundamental problem remains in these networks without an adequate answer: how to go from a box of sensor nodes to a functioning network. This is known as network set-up, configuration, commissioning, or bootstrapping. Bootstrapping is complete when a node is able to securely exchange information with another node over the network. Bootstrapping solves problems such as what channel to run on and how to exchange initial encryption keys.

The difficulty in finding an acceptable solution stems from the wide range of requirements, as typical considerations for wireless sensor networks include: deployment, if it is a one-time setup or a network which is continually expanding; mobility, if the nodes can move about requiring the network to continuously reform; cost of nodes; size of nodes; computational resources of nodes; power source of nodes; network topology, normally either a star or mesh network; management, be it a centrally managed infrastructure or ad-hoc network; size of network; security, which could include key exchange and cryptographic algorithms to use; and Quality of Service (QoS) requirements [1].

In light of such a wide range of sometimes conflicting requirements, it may seem hopeless to devise a universal method of configuring nodes. From the previous considerations a set of generic problems will be put forth. Then these problems will be distilled into the base requirements. Existing similar solutions will then be compared to these base requirements, where suitable and unsuitable solutions are discussed. Finally the successful methods will be summarized with a discussion of implementing them.

II. Problems And Requirements

A. Network Partitions

During network setup, two separate networks may be created instead of a single unified network. These two networks have no knowledge of each other, and hence joining them together is a bootstrapping problem.

An example would be a RF remote-control. If a consumer purchases a TV and DVD, they will each come with a remote control which may be initially coded to work with their respective device. When the user wishes to use one remote to control both devices, the two separate (TV and DVD) networks must merge.

1) Requirement: Merging

The bootstrapping system must work regardless of the current node's status. If a user wishes to join two nodes which are already on a network, the system should allow the user to merge networks together.

B. Ad-Hoc and Networks without Central Authority

Many networks will either run without a central authority, or initially be created without a central authority. In a WSN it may be non-obvious which node is the central authority, and some network topologies have no central authority. Thus the protocol for node configuration must be flexible enough to run from almost any two nodes.

1) Requirement: Distribute Trust

Many topologies of wireless networks will be implemented, ranging from ad-hoc networks to centrally administered networks. The bootstrapping protocol must work with both types of networks; meaning end nodes can authorize other end nodes. Management policies may disable such abilities; end nodes which an attacker has physical access to for instance should not be allowed to authorize new nodes.

C. Node mobility

Node mobility naturally presents a problem in most networks. The low cost and size of the nodes means they are easily damaged or stolen, with new nodes brought in to replace the lost nodes. Any network configuration must provide a quick manner of adding a new node into the network and recovering old information [2], [5].

1) Requirement: Support Node Mobility

A network may be dissolved, and the nodes re-used in another network. It thus must be possible to make a node drop off the previous network and join another network.

D. Resource Constraints

Undoubtedly the single largest obstacle is resource constraints. This includes the fact that nodes must be cheap and consume a minimal power. These two requirements conspire to limit the nodes human interface capability, processing power, physical size, and network connectivity.

1) Requirement: Support Low Resource Nodes

The method must be flexible; a minimum implementation must be deployable on very constrained devices.

E. Security

Any wireless sensor network must have security, due to the extreme ease of which passive or active attacks can be attempted on the networks. Most cases require simple authentication: preventing the neighbors from controlling your TV for example. The security problem in bootstrapping is primarily concerned with how end users are expected to easily secure their networks, and not security in use during network operation.

1) Requirement: Support Security

The method selected must also work with the security present in the network. At minimum it must provide a method of securely establishing a link between two nodes. Through this channel security information can be exchanged such as encryption keys. This secure channel is only for bootstrapping; other security layers take over during normal operation.

III. Existing Solutions

This section will provide a description of several existing solutions.

A. Device Label / Certificate

Using a label to hold a shared secret is described in the Wi-Fi Protected Setup (WPS) specification [3]. Devices have labels with a number printed on them (the PIN), this can be used to form a shared secret between two devices. To authorize a cell phone on a Wi-Fi network for example, the user enters the PIN found on the underside of the access point. Later when they bring home a printer, they use the cell phone to enter the PIN from the underside of the printer. This again forms a shared secret and the device can be securely authorized.

B. Button-Press Algorithm

The button-press algorithm is very easy for an ‘end user’ to understand and used in several standards. The user sets two devices in a special ‘join’ mode where they then discover each other and join up. This special mode could be entered by a physical or virtual button. Both WPS and Bluetooth Secure Simple Pairing (SSP) use a form of this button-press method [3], [4].

C. Resurrecting Duckling Method

A method proposed by Stajano and Anderson is based on biological origins [2]. The idea is that most devices should associate with the first node they see; the example being how when a duckling hatches it associates with their mother instinctively. Requiring a node to associate with another network is simply a matter of ‘killing’ the node, and upon being ‘resurrected’ it can be made to associate with the new network.

The act of killing the node can either be done through physical access to the node, or requiring the original mother duck or some other master [5].

D. OOB: User Comparison

Bluetooth extends the ‘Just Works’ method to also include a numeric comparison. Here the devices authenticate as in ‘Just Works’, but they also compare a number presented on the display of both devices. This comparison is used to prevent a Man In The Middle (MITM) attack which ‘Just Works’ has no defenses against[4].

E. OOB: Optical Exchange

Optical exchange of information uses some optical communication as an OOB method. The reason for using optical could be a requirement of the device, such as medical, or because the device already has an optical channel built in.

The method proposed in [6] is aimed at the medical market. Nodes are equipped with an IR sensor, and a special IR pen can download information to the nodes. For medical applications additional requirements are imposed beyond security; people would generally object to having a USB port installed in their arm to allow node management.

Roman and Lopez demonstrate a method of using a normal LED and light sensor instead of IR devices called KeyLED[7]. An extension of this is a solution presented by Long & Durham, using the human as the light sensor [8].

F. OOB: USB

Using USB as an OOB communication method is suggested in WPS [8]. Many nodes may already have a USB interface, and using it would be a natural choice to avoid requiring

additional hardware. The USB interface would also allow more advanced configuration, such as plugging a node directly into a computer for either configuration or use. USB On The Go (OTG) could allow plugging two nodes together.

G. OOB: Near Field Communication (NFC)

Using a NFC token is another OOB channel to transfer security information. In this case a token is touched to the master node, and then touched to the end device. The token transfers information used to securely gain access to the network. For low-powered and low-cost devices the NFC token may be integrated directly into the end node. Instead of touching the NFC token to the master node, you instead touch the end node to the master node. An NFC token can be powered from the reader, making this a no-power solution.

H. OOB: Simple Physical Link

Using USB is one form of a physical link, with well-defined standards and physical connectors. However almost every node in existence supports one other physical link: the physical link used to initially download firmware and settings. These tend to be application and vendor-specific however.

IV. Analysis of Solutions Against Requirements

With a set of requirements which a bootstrapping method must follow, the examples can then be compared to those requirements. The requirements correspond with those defined in section III.

A. Merging Multiple Networks

None of the protocols discussed in section III specifically deal with merging networks. Any method which allows a user to command that two nodes should join together has the potential to support node merging.

B. Distribute Trust

All the methods could be adapted to work with a distributed trust system; possibly with the exception of infrastructure mode.

C. Support Mobile Nodes

All of the methods outlined have a method of adding and removing nodes from a network.

D. Minimal Resource Usage

Resource usage remains very important in WSN. These include computational, cost, and power requirements. Computational requirements eliminate the direct application of many standards such as Wi-Fi and Bluetooth, however the methods in those standards could be modified to use less resource-intensive computations.

Cost and power requirements by comparison, will form requirements which are much more difficult to pass.

Methods which require a unique label printed on the node enforce on every node a minimum size requirement and the cost of matching unique labels to preprogrammed information. Any OOB interface not used in node function will naturally require additional hardware, such as IR and NFC. This hardware may only be used once in the node's lifetime, making justification of the hardware's cost and size difficult.

E. Security

Security provides a wide range of issues which dictate the suitability of various protocols. There are many attacks possible on a wireless network in operation; the narrow specification of securing during bootstrapping however means they are out of scope for this paper [5]. Attacks during set-up include passive listening, interception attacks, timing attacks, and physical attacks. Each of these will be considered in sequence.

1) Passive Listening

The easiest attack on a RF network is passive listening. An attacker with very low-cost hardware can perform this attack.

Securing the initial exchange of data requires two nodes to have a shared secret which is unknown to the attacker. The easiest method to transfer this shared secret is with an OOB method, such as the device PIN, a NFC token, optical transmission, or key pre-distribution [3]. The second method is to use a public-key exchange based on the Diffie-Hellman method [9], which can be implemented on constrained devices with the proper algorithms [10].

Any method which proposes an exchange based on an unsecure RF link, and simply hoping that during a brief period no attacker is listening, should immediately be rejected. Similarly any node so constrained it is incapable of performing a cryptographically secure exchange should be disallowed from joining the network over the air.

2) Interception Attacks

Interception attacks occur when an attacker is able to insert themselves between one node and another. The Man In The Middle (MITM) attack is one of these, where two end nodes think they are directly talking to each other, but are actually talking through an attacker [4].

3) Timing Attacks

Timing attacks are ones which simply require an attacker to be present during a certain time. As an example consider the WPS button-press protocol. If the user initializes the protocol by pressing the 'Join' button on an access point, there is some delay before they press the button on the end node. During this time an attacker could simply run the join protocol on another end node. Any method which does not definitively

confirm with the user which two devices the user wishes to join could be attacked in this manner.

4) Physical Attacks

Physical attacks include a range of attacks, and many approaches must be considered. It is assumed that an attacker may take possession of an end node, and such an action must not compromise the network. A very simple printed label for example may provide too much information to an attacker. Thus a printed PIN must be carefully designed, possibly using a public-key method, if there is a risk nodes could fall into the hands of attackers.

V. Proposed Solutions

Selecting the proper method depends on the exact requirements, as each solution has advantages and disadvantages. Of the presented solutions four were selected as a suitable method in the widest range of situations: a push-button method, a device PIN or label, wireless OOB transmission, and physical OOB transmission.

A. Push Button

The push-button method here is a combination of multiple configuration methods [2], [3], [4], [8]. The minimum requirements to run this protocol are an LED and push-button. Upon button press, the node enters 'Configuration' mode. If the node is un-configured, it begins a scan for networks to which it can join. Once a network is found it joins that network, using a Diffie-Hellman exchange to protect from passive eavesdropping. Before completing the join request, both nodes blink out a unique pattern on their LED dependant on the shared secret created from the Diffie-Hellman exchange.

The user can then compare that both LEDs are blinking on and off at the same time, and press the button on each node to confirm the join request. This provides some protection to a MITM attack, similar to Bluetooth Secure Simple Pairing where a user compares two numbers on each device [4]. Even if the user does not carefully confirm the blink pattern they will at minimum confirm that the two proper devices are blinking.

If a device is already configured when the button is pressed, there are several options. In ad-hoc networks pressing the button results in the node listening for an un-configured node to connect. Once this un-configured node attaches it will be added to the network. A centrally managed network would not allow any end node to authorize a new node on the network, meaning the button press would be ignored. Holding down the button would result in the node configuration being cleared, allowing the node to leave its current network.

Network merging is also accomplished through the button press protocol. Whenever the button is pressed on an already-configured node, it does a quick scan for any other nodes which are advertising a network to join. If found this may mean the user wishes the two networks to join together. In this way if two nodes, both already configured, have their button pressed, the node which was activated first will do a scan looking for other advertising nodes. Finding none it will then proceed to advertise its own network. By the time the user has moved to the second node and pressed the button, the first node will already be advertising the network. Thus when the second node does a quick scan it will pick up that an already configured nodes exist in the same space.

When running the merge, it is important the user is aware which nodes will be merging. All nodes equipped with the 'join' LED will flash in unison on both networks. The user again must press the button on each node to confirm this. If this was not intended, the user could hold down the button on one node to clear the node's memory, and thus disconnect the node from the network. Running the protocol again would result in the node associating with the new network only, and not merging both networks.

B. Device PIN / Label / Certificate

As previously mentioned the printed label can be used to provide an easy method of distributing a shared secret, though the discussion also focused on possible attacks. There are a variety of cases where this security is sufficient though, and the possible attacks on the nodes are not realistic.

Such examples would include nodes which when broken will simply be discarded. Examples could include smart dust configurations, where many nodes need to be deployed. As each node is deployed a 2D barcode on it is scanned and a central authority gives the node network access.

C. Non-Contact Out Of Band

Certain situations require a non-contact configuration solution. Prime examples would include the medical field and intrinsically safe devices. For these devices they do not need ad-hoc association, hence embedding NFC tokens in them results in an ideal solution.

D. Physical Out Of Band

Using a physical connection assures a secure connector as any attacker would have to have access to this line. The physical line also allows fast communication, minimum power consumption, minimal cost, and minimal size.

Power consumption is minimized as no transceiver is needed as would be with IR, an NFC reader, or certain other mediums. In addition the physical interface itself could power the end node during the configuration process.

Cost is minimized as the physical interface would require a minimum of external components. The physical interface need not have a single defined connector. Multiple accepted

connectors means that the simplest connector would simply be pads on a PCB. The mating side would be spring-loaded pins which make a connection. Slightly larger nodes could use easily mateable connections such as 2.5mm 'tip, ring, sleeve' plugs. The largest nodes could use a connector which resists harsh environments.

Finally size is also minimized from the lack of external parts required. Almost all nodes will require the physical connection anyway for device programming and configuration; standardizing this connection is the logical method of minimizing waste.

A physical line could then be used in almost all network configurations. A simple home network might ship in a blister back consisting of three nodes and a small pen-sized device. The user simply plugs the pen-sized device into each node they want on the same network; when they want to add nodes later they just plug that same pen into new nodes. Institutional installations might bring a small laptop along during installation. A USB adapter allows nodes to be plugged in, with specific keys loaded and node-specific configuration such as serial numbers set in the field.

VI. Conclusion

Bootstrapping a network from a box of nodes to a working network is a problem without one perfect solution. The wide requirements of different network topologies prevent a single solution from working perfectly in every case. By understanding the base problems however, a subset of the many possible solutions can be selected that works in most cases.

To achieve interoperability between nodes and vendors, this standard must also be agreed upon. Even just agreeing on a certain number of standards is not enough to ensure interoperability. Every node will not be able to implement every standard; most may only be able to implement one or two. Hence the 'lowest common denominator' needs to be identified.

The lowest common denominator is the simple physical link, and should be the only required implementation. On small nodes it would not require precious extra resources; yet higher performance nodes do not suffer a loss of security by being forced to implement this method.

The analysis of existing bootstrapping suggestions and standards shows much promise in reaching an ideal method, considering the often conflicting requirements. This paper is not designed to be implemented as a standard; rather it is hoped to reach towards a goal of a unified bootstrapping method usable across as many environments as the radio network itself.

References

[1] Romer, K.; Mattern, F., "The design space of wireless sensor networks," *Wireless Communications, IEEE* , vol.11, no.6, pp. 54-61, Dec. 2004

[2] Stajano, F.; Anderson, R., "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks," In *Proceedings of the 7th International Workshop on Security Protocols*. LNCS. vol.1796, pp 172-194, 1999.

[3] Wi-Fi Alliance, "Wi-Fi Protected Setup Specification v1.0", 2007

[4] Bluetooth Special Interest Group, "Simple Pairing Whitepaper," Revision V10r00, Aug 2006.

[5] Stajano, F., "Security Issues in Ubiquitous Computing," *Handbook of Ambient Intelligence and Smart Environments*, part III., pp 281-314, 1st ed., 2009.

[6] Baldus, H.; Klabunde, K.; Müsch G., "Reliable Set-Up of Medical Body Sensor Networks," *Wireless Sensor Networks*, LNCS, vol.2920, pp. 353-363, 2004

[7] Roman, R.; Lopez, J., "KeyLED - transmitting sensitive data over out-of-band channels in wireless sensor networks," *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on* , vol., no., pp.796-801, Sept. 29 2008-Oct. 2 2008

[8] Men Long; Durham, D., "Human Perceivable Authentication: An Economical Solution for Security Associations in Short-Distance Wireless Networking," *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on* , vol., no., pp.257-264, 13-16 Aug. 2007

[9] Diffie W., Hellman M.E., "New Directions in Cryptography", *IEEE Transaction on Information Theory*, Vol. IT22, No. 6, Nov 1976.

[10] Leif Uhsadel, Axel Poschmann, and Christof Paar. Enabling Full-Size Public-key Algorithms on 8-bit Sensor Nodes. In Proceedings of ESAS 2007, volume 4572 of LNCS, pages 73--86, 2007.