Network working group
Internet Draft
January 2002
Expires: July 2002

Stephen R. Hanna, Sun Microsystems, Inc.

draft-hanna-zeroconf-seccfg-00.txt

Configuring Security Parameters in Small Devices

Status of this Memo

Abstract

Before a device is installed into a secure network, certain security parameters (such as keys) must be configured. This document describes several techniques for establishing these parameters and analyzes their advantages and disadvantages, considering especially their suitability for inexpensive devices and inexperienced operators.

1. Introduction

The IETF's Zero Configuration Networking working group is working on protocols that allow devices to form a network with little or no configuration. Unfortunately, securing such a network is problematic. All proposed security techniques require some configuration. Otherwise, your new lamp won't know that it should respond to your commands and not your neighbors'.

The challenge is to make this configuration as easy as possible and keep the per-device cost low. This document describes several techniques for establishing security parameters and analyzes their advantages and disadvantages, especially their suitability for inexpensive devices and inexperienced operators.

This document does not describe the exact set of security parameters to be established or how those parameters should be used in a zeroconf environment. Another document will tackle that problem. But a simple example of such parameters would be a single symmetric key shared by all trusted devices on the network and used for encrypting and authenticating transmissions between those devices.

2. Evaluation Criteria

Several criteria are examined for each configuration technique. For each criterion, I will rate the schemes on a 1-5 scale with 1 being unacceptable and 5 being excellent.

No perfect solution has been found. Each technique provides different tradeoffs among the various criteria.

2.1. Device Cost

Solutions should increase the cost of the device as little as possible. Cost is a crucial criterion for consumer goods. My goal is to keep incremental cost so low that anyone who creates a device with zeroconf networking abilities will include security.

2.2. Ease of Use

Solutions should be as easy for the user as possible. A slow, complex, or cumbersome process is a barrier to use.

2.3. Security

Solutions should be as secure as possible. Security is the point. See section 3 for a threat analysis.

2.4. Flexibility

Solutions should be flexible, so that they can work with many different kinds of devices. For instance, a solution that requires the device to display a number won't work if the device doesn't have a display.

3. Threat Analysis

The goal is to easily configure a device so that it can communicate securely with other devices on a network. I will only analyze configuration mechanisms, not the protocols used to communicate among devices. But it is still important to have a clear understanding of the threats against which these configuration mechanisms are designed to protect. I will examine attacker's goals, motivations, and abilities.

3.1. Attacker Goals and Motivations

Attackers may want to read messages sent by the device, such as audio signals or other monitoring data from sensors. They may also want to control the device by sending it commands, such as the ability to open doors or windows, turn lights on and off, etc. Or they may just want to detect homes that have wireless networks and therefore might be good targets for burglary.

Attackers' motivations may include curiosity, mischief-making, malicious destruction, or financial gain.

3.2. Ability to Read and Modify Network Data

I assume that attackers can read or modify data sent over the network and inject arbitrary messages into the network. This is consistent with networks commonly used in zero configuration environments, such as a wireless or power line network. Layer 2 encryption can be used to secure such networks and should be employed when available. However, some layer 2 encryption systems are easily bypassed or broken. Also, some networks do not offer level 2 encryption and a zero configuration network can span multiple layer 2 networks, thus providing a vulnerability.

Even a network completely protected against outside participation can be vulnerable to these attacks if some of the devices on the network become compromised or act as gateways for outside networks. In general, it's best to assume that attackers have access to the network.

3.3. Physical Security

In general, I assume that attackers are not able to violate a device's physical security. With inexpensive devices, even a brief period of unmonitored physical access will generally allow a skilled attacker to compromise a device or replace it with an identical, compromised unit.

In the zero configuration environment, these assumptions may not hold. Neighbors, workmen, and other partially trusted guests are often welcomed into one's home with little supervision. Even roommates or family members can become adversaries at times, perhaps resulting in "insider attacks". Device security cannot offer a great deal of protection against such problems. However, some of the configuration systems noted below provide a bit of protection against physical attacks, especially against unskilled attackers. I will note such protection where present.

3.4. Device Vulnerabilities

The attacker may be able to take advantage of bugs and

vulnerabilities in the device software or hardware to compromise the device (for instance, sending an invalid message that causes a buffer overflow and allows the attacker to execute arbitrary code). Configuration solutions cannot offer much protection against such vulnerabilities.

Device manufacturers often include backdoors that their support staff can use to help customers gain access. Such backdoors can represent a serious security hole. If they decide that backdoors are required, manufacturers should design them so that they require physical access to the device.

3.5. Denial Of Service

An attacker who can send data on the network can probably easily jam the network with packets (or just with noise). Configuration techniques can't provide much protection against such attacks. However, this sort of active attack may make it easier to find an attacker.

3.6. Configuration Attack

Most people probably won't ever enable security on their device. An attacker can mount a clever denial of service attack by configuring security on the device (or reconfiguring it, if it's already enabled) so that the user can't access the device. To prevent such attacks, security should be hard to enable (requiring physical access and perhaps a password shipped with the device) and easy to disable (if you have physical access to the device).

Of course, this will make it easy for attackers with physical access to the device to disable security. But a home controller can quickly detect this and notify the owner. Requiring physical access to disable security (pressing a button) is a reasonable compromise. After all, device manufacturers will not include security features if they are likely to increase support costs.

3.7. Traffic Analysis

Various attacks based on traffic analysis may be possible. One of the more obvious is for a burglar to drive around with a wireless networking antenna, detecting wireless network traffic to decide which homes are likely to have computers inside. Possible countermeasures could include using wired networking to make detection more difficult. Installing a monitored security alarm and purchasing sufficient theft insurance would also be prudent. In any case, this has little to do with configuration.

3.8. Summary

Device security typically offers little protection against

physical attacks and insider attacks. However, some protection can be offered by employing a controller that monitors devices and notifies the owner when devices change.

Since device security can be used to lock someone out of their devices, it should be easy to disable device security if you have physical access to the device. The home controller should detect and log such changes, in case they were unauthorized.

The primary area where device security can be effective is against attacks that are mounted remotely (through the network). Such attacks can be mounted with little risk or cost to the attacker and without raising the suspicions of the attacked party, especially when a wireless network is employed. The configuration techniques described in this document provide a way to configure security parameters (such as keys) that may be employed in protocols to protect against remote attacks.

4. Security Configuration Mechanisms

On conventional systems, initial security parameters such as trusted keys and security policies are usually typed in with a keyboard or loaded from floppy disk or other removable storage device such as a CD-ROM or smart card. Once those initial parameters have been established, they can be used to securely deliver other things such as software and updated security parameters.

Many consumer devices don't have a keyboard or removable storage device. Adding these things would increase the cost of the device too much. Some devices have a limited keyboard and display, but entering a long key into my microwave oven isn't my idea of fun!

Here are some other ways that security parameters can be configured into consumer devices.

4.1. Secret stored in device during manufacturing

With this solution, a cryptographic secret is stored in the device during manufacturing. This secret generally cannot be changed. The secret is also printed as a bar code, which is sealed in a tamper-evident manner and shipped with the device. When the device is purchased, the bar code is scanned into a home controller or other security manager. The security manager can now use this shared secret to send the security parameters to the device over an untrusted network (encrypting the data with the secret).

The secret can be transferred to the security manager in many other ways. It can be printed in a human readable format and typed on a keyboard, stored on and read from a mag stripe card or a floppy disk, etc. In general, I will say that it is transported to the security manager via a "secure transport mechanism". Including a

keyboard, barcode reader, or removable storage device in the security manager shouldn't be a big cost problem. You only need one security manager per home and it will probably have a keyboard and display to interact with the user, anyway.

The secret could be transmitted from the device to the security manager over a secure network, such as a dedicated wire, a proximity network (like infrared or Bluetooth), or by touching electrical contacts. However, if a secure network is available, it is usually better to generate the secret on the fly (as describe in section 4.2). This reduces manufacturing cost and eliminates the risk of the user losing the secret. Therefore, this option is not analyzed any further.

It is not good to just print the secret on the bottom of the device, or a casual visitor can read it without detection. Putting the secret on a storage mechanism (like paper) that is included with the device allows the user to keep this copy of the secret in a safe place, away from prying eyes.

Making the secure transport mechanism tamper-evident (by wrapping it with a printed piece of plastic or using a sealed piece of paper) makes it less likely that someone will copy the secret before the user receives it, without their knowledge. This attack seems unlikely, especially if the user buys the device from a large store. Therefore, I conclude that tamper-resistance should only be used if it doesn't add much to the cost or decrease ease of use much.

Another way to get the secret to the security manager is to have the security manager download the secret from a central server by providing the device's serial number or some other identifier. Unfortunately, such identifiers are usually not secret (or there would be no need to download the secret!). So it would be easy for an unauthorized party to find out the serial number and download the secret. So this isn't a great answer, either.

In order to protect against configuration attacks (where someone turns on security and the owner doesn't know how to use it), there should be a switch, button, or other control on the device that enables and disables security. It should be shipped with security off, since most people won't want to use security.

Someone with physical access to the device could use the switch to disable security, but the security manager should be able to detect this and notify the owner. Why not allow security to be enabled over the network? Because that means you can mount a configuration attack without physical access.

4.1.1. Evaluation

Cost

The device will need a place to store the secret. However, it would

already need a place to store the security parameters, so this probably will not add much to cost.

Including a switch to enable and disable security may add some significant cost (up to $1). Devices that already have a keyboard can use a special combination of keys for this, eliminating any incremental cost.

Manufacturing cost will increase, due to the need to generate the secret, store it on the device, and place it on the secure transport mechanism. The cost of the secure transport mechanism must be considered, as well.

And there must be a security manager that knows all of the device secrets and supports the secure transport mechanism (via a keyboard, a barcode reader, or some similar mechanism). However, this cost may be spread across many devices. Also, this device can serve other functions as well, such as providing a UI for controlling and configuring devices.

The incremental cost per device will vary. For simple devices, the cost of the security switch may be significant. But increased manufacturing costs will probably be the greatest factor.

I'll rate this scheme a 3 for cost.

Ease of Use

Scanning the secret into the security manager should be pretty easy. But what if the user loses the slip of paper first? Or what if their security manager breaks? They need to find the slip of paper to scan into the new security manager. In many households, the slip will be lost quickly. Flipping the switch to enable security is a minor irritant compared to this problem.

I'll rate this scheme a 2 for ease of use.

Security

What if someone nasty learns the secret? They might see the slip of paper, take the security maneger, or some such. There's no way to change the secret. So they will have total control over the device. And the device's owner may not even know about this.

I'll rate this scheme a 2 for security.

Flexibility

This scheme requires a switch or similar control to enable and disable security (unless you're willing to live with configuration attacks). It will be inconvenient for devices that are hard to access, such as a ceiling light. But enabling security should be a one-time event. Overall, this scheme gets high marks for flexibility. It should work

with almost any kind of device.

I'll rate this scheme a 4 for flexibility.

4.2. Secret established over a secure network

When the user wants to enable security, she connects the device to a security manager via a secure network (such as a wire directly connecting the two). Then she presses a button on the device, which generates a new secret and sends it to the security manager over the network. Now the device can be removed from the secure network and security parameters can be downloaded securely from the security manager, using the secret. Security can be disabled by holding down the button for a longer period of time.

As a further alternative, the secret can be generated by the security manager instead of the device. This makes more sense, since the security manager will probably include a good source of entropy. The device can send a secret request when the security button is depressed and the security manager can respond by sending a new secret. This also means that security could be disabled by disconnecting the device from the secure network, pressing the button to trigger a security request, and noting that no response was received.

What kind of secure network can be used here? Many things will work: a dedicated wire (like a USB cable), a proximity network (like infrared or Bluetooth), or physical contact by touching electrical contacts. Probably the simplest thing to do is to use power line networking. Most devices have a power plug and the security manager can have a special socket that's isolated from the rest of the power grid. Devices that don't have a power plug may have a recharging stand that has a power plug and could relay communications to the device.

There are a few obvious alternatives to the secure network. The security manager could display the secret after generating it and the user could type that value into the device's keyboard. But that's much too painful for the user and it requires the device to have a keyboard (and a display, to check for typos). If the device generates the secret and displays it, then the device only needs a display but the user still has to type the secret (which will typically be dozens of letters and numbers). The device could print the secret as a barcode or store it on a floppy disk or other storage device, but including a printer or storage device in every toaster and light switch is much too expensive.

4.2.1. Evaluation

Cost

The cost of storing the secret and the cost of the security button should be the same as with the previous scheme.

The incremental manufacturing cost of the last scheme isn't required for this one, since there's no need to generate a secret, print it, etc.

There may be extra cost for including the secure network. However, many devices that support zero configuration networking will probably already include support for power line networking, which means that no extra cost will be incurred.

I'll rate this scheme a 4 for cost.

Ease of Use

Connecting the device to the security manager and pressing a button is pretty easy.

I'll rate this scheme a 4 for ease of use.

Security

If a secret is compromised, it's easy to generate a new one with this system. As with the previous system, anyone with physical access to the device can disable security with the press of a button. But that's a good thing, since it makes it easy to recover from configuration attacks. And it should be easy for the security manager to notice if security is disabled on a device.

I'll rate this scheme a 4 for security.

Flexibility

With a device that doesn't have a power plug (like a light switch), it may be difficult to establish a secure network. A secondary cable could be used, but this will require extra cost and it may be difficult to agree on a standard connector and cable type.

I'll rate this scheme a 3 for flexibility.

4.3. Secret established over an insecure network

Wouldn't it be nice if the device and the security manager could establish a shared secret over the normal zeroconf network? Then there would be no need to use a secure network, as in the last scheme.

Actually, there is a way to do this, even if nasty folks are listening in. It's a Diffie-Hellman exchange. Here's a simplified description. A large prime number p and a generator g are chosen in advance and are well-known. The device and security manager each choose a random number between 1 and p-2. Call the device's

random number a and the security manager's number b. The device computes g^a mod p and sends this value to the security manager. The security manager computes g^b mod p and sends this value to the device. Now each of them can compute g^(ab) mod p. And no eavesdropper can determine g^(ab) mod p without a *lot* of work.

The problem with this system is that the device and the security manager can't be sure they're talking to each other. There might be a "man-in-the-middle" who has established a different shared secret with each of them. Or maybe the interloper just impersonated the security manager and the security manager wasn't aware of any exchange? Since the network is not secure, this sort of thing can happen.

One way to overcome this problem is to have the device and the security manager hash the newly-established secret and display a few bits of this hash. The user can then compare the displayed values. If they match, then the secret has been established properly.

This system has several problems. Most important, it requires too much work from the user. Also, it requires a display on the device. This isn't practical for light switches and other small or cheap devices. Finally, computing g^a mod p and g^(ab) mod p in a few seconds requires lots of compute power (or a p that's too small to be secure). And the device must have a good-quality random number generator. Much too expensive for a cheap device.

4.3.1. Evaluation

Cost

> Including a display, high-quality random number generator, and serious computing power in the device will increase cost of goods substantially (probably $10 or more).
>
> I'll rate this scheme a 1 for cost.

Ease of Use

> Copying down a long string of numbers and letters and comparing them to another set of numbers and letters isn't easy or pleasant.
>
> I'll rate this scheme a 1 for ease of use.

Security

> If the user fails to compare the secrets, a man-in-the-middle attack is possible. The user has to do this for each device.
>
> I'll rate this scheme a 2 for security.

Flexibility

Small devices may not have space for a display.

I'll rate this scheme a 3 for flexibility.

4.4. Other systems

Many other solutions have been considered and rejected. Here's a brief description of some of these solutions.

The user could choose a secret and enter it into the device and the security manager. But users don't generally choose good cryptographic keys (which must be very long and truly random numbers). An attacker with a computer could find a user-chosen key in a fraction of a second.

The security manager could choose a secret and then have the user enter it into a keypad on the device. But this would not be easy for the user and it would require a keypad on the device (and a display, to check for errors).

The device and/or the security manager could have a public-private key pair (and perhaps a certificate). But then I must securely configure the public key or certificate identifier for one into the other. This is similar to the problem of establishing a shared secret, above. And public key cryptography requires much more CPU power than symmetric cryptography.

5. Conclusions

5.1. Configuring Security Parameters

Security parameters can be safely configured with the systems described above. Several of these systems are practical and add little or no extra cost per device. Some user configuration is required, but that seems to be inevitable.

I recommend that a standard be created (maybe by the IETF, maybe by someone else) that describes the first two schemes listed above (Secret stored in device during manufacturing and Secret established over a secure network) and develops them into a form that can be incorporated into shipping products in an interoperable manner.

Most devices should probably use the secure network technique, since this is easier for the user and less error-prone. Devices that do not have a power plug may want to use the secret stored during manufacturing technique.

Certainly, there may be other good techniques. I welcome ideas from others. Please send them to my email address, listed below.

5.2. Zeroconf Security

Currently, each of the zeroconf protocol specifications has its own security system. Each of these systems requires configuration and complexity. This is not practical. I recommend that the zeroconf working group create a security specification that either explains how these several security systems can be automatically bootstrapped using the mechanisms described above or (preferably) replaces these security mechanisms with a simpler system such as IPsec with a single group key shared by all authorized members of the zeroconf network.

6. Acknowledgments

I would like to thank Erik Guttman, Radia Perlman, and Aiden Williams for useful discussions on these topics.

Ross Anderson's Resurrecting Duckling paper [1] describes the need to "imprint" a device with security parameters and describes one solution (electrical contact).

7. Security Considerations

This document is full of security analysis and proposed security solutions.

8. References

[1] Stajano, F. and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks", from Security Protocols: 7th International Workshop Proceedings, Lecture Notes in Computer Science, 1999, Springer-Verlag.

9. Authors' Addresses

Stephen R. Hanna
Sun Microsystems, Inc.
One Network Drive
Burlington, MA 01803 USA
Phone: +1.781.442.0166
Email: steve.hanna@sun.com

10. Intellectual Property Statement