Protecting Domestic Power-line Communications

Richard Newman[1], Sherman Gavette[2], Larry Yonge[3], Ross Anderson[4]
University of Florida
[1]Computer and Information Science and Engineering Department
PO Box 116120 Gainesville, FL 32611-6120 USA
nemo@cise.ufl.edu
[2]Sharp Labs, USA
[3]Intellon Corporation, USA
[4]Cambridge University, UK

***Abstract* – In this paper we describe the protection goals and mechanisms in HomePlug AV, a next-generation power-line communications standard. This is a fascinating case-history in security usability. There are also novel protocol issues; interactions with mechanisms at other layers; and opportunities for both researchers and third-party vendors to build on the mechanisms provided. The central problem – being sure whether a device being enrolled in the network is the device you think, not a similar one nearby – is not well solved by conventional mechanisms such as public-key infrastructures, but appears to require either very old-fashioned or very novel approaches.**

Categories and Subject Descriptors – K.6.5 [Security and Protection] – Authentication

General Terms – Security, Human Factors.

Keywords – authentication, power-line communication, security.

I. INTRODUCTION©

Low-bandwidth power-line communications, such as X10 [1], have existed for many years. Since 2000, 14-Mbps HomePlug 1.0 equipment has been commercially available for in-home power-line communication [2]. This technology provides data communications aimed at computers and gaming. The HomePlug consortium (sponsored by Cisco, Comcast, Earthlink, GE, Intel, Motorola, Radio Shack, Sharp and Sony) is now developing an improved, 150 Mbps standard called HomePlug AV for multimedia applications [3,4,5,6]. The goal is that all kinds of electronic equipment should be able to communicate within the home (or office) via the power mains. A consumer buying a personal video recorder will simply plug it into the mains, whereupon it will discover the TV, the set-top box, and other relevant equipment. The devices will form a logical network without the need for additional physical wiring.

This raises interesting and important questions of security and reliability. Many customers live in apartments and other buildings that share power lines, and so signals can cross property boundaries just as wireless signals can. If I bring home a video recorder and plug it in, how can I be sure that it connects to my home network rather than my neighbor's? There may be other boundaries at an even finer granularity. For example, students occupying a shared house might want to have one network each, and adolescents might want bedroom networks distinct from the general network in their parents' house. So we have to support multiple virtual networks. However, security management in traditional systems is beyond the average person's patience. The majority of home wireless LANs go unprotected [7].

Power-line communications are similar, from the security viewpoint, with short-range radio communications such as wireless LANs, Bluetooth and UWB. There are three main differences that make the security design exercise different and instructive. First, while short-range radio is inherently range-limited, power-line networks can become unmanageably large. If all the devices in a large apartment block are allowed to assemble themselves into a single network, the performance drops significantly. This phenomenon, known as 'The Borg', means that networks may have to be

partitioned into logical networks for performance reasons, even if security is not an issue.

Second, power-line networking is aimed at a very wide range of consumer electronic devices, from PCs and DSL routers down to devices such as fire alarm sensors and loudspeakers. Not all of these devices have CPUs capable of public-key cryptography, and not all have rich user interfaces: some may have no more than a reset button.

Third, the physical layer provided by the modulation scheme in HomePlug AV [3,4,5,6] can provide a certain amount of assurance even in the absence of cryptography. It has basically two modes. In broadcast mode, the bit rate is low but if two stations transmit simultaneously, this is likely to be detected. Normal mode is point-to-point and uses a much higher bit rate. In order to achieve this, tone maps (bit loading choices per carrier) must be adaptively selected for each direction of communication on each virtual link. This makes wiretapping fairly difficult, in ways that we will describe more fully below.

This paper describes the issues involved in (and explores other possible approaches to) designing the security layer for this protocol standard. It must not only have satisfactory security characteristics, but also support desirable experiences for a wide range of users. Before proceeding with the details of security requirements and architecture, we first give a basic introduction to power-line communications, and to the emerging HomePlug AV standard.

II. In-home Power-line Communications and HomePlug AV

The power mains in homes and small businesses are inherently a broadcast medium, with frequency-selective attenuation dependent on where the transmitter and the receiver attach. Attenuation is high for all frequencies, and there is much noise of various types, so carrier detection is difficult, and collision detection is even harder. Hence, earlier systems used Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) for access to the medium, as in IEEE 802.11. (CSMA is "listen before talk," and in the absence of reliable collision detection, stations only attempt to access the medium probabilistically.) Virtual Carrier Sense, based on information supplied in the robustly broadcast frame control field, is used to inform medium access decisions. Since the frame control must be very reliable, it is heavily coded and is inefficient.

To make the most of the channel, each pair of communicating stations adapts the bit-loaded Orthogonal Frequency Division Multiplexing (OFDM) modulation according to the current channel characteristics [8]. In HomePlug AV, this means choosing one of eight possible modulation rates (from none to 10 bits per symbol per carrier) for each of 917 carriers. This modulation information, along with the forward error correction coding rate (1/2 or 16/21) and the guard interval duration (three choices) constitutes the tone map. This tone map is set by the receiver and then used by the sender to transmit the data payload. An attacker might be able to measure which of the roughly 2^{2753} possible tone maps is in use on a particular link, but even knowing the tone map, demodulation by a station other than the intended recipient is problematic. The modulation rate for each carrier is adapted to be very close to the maximum rate possible given the signal to noise ratio. While not impossible, interception of the data payload is a significant challenge, which we discuss below.

Tone-mapped communication requires that sender and receiver agree on the tone maps, which in turn requires some initial communication. Two broadcast tone maps are defined for this purpose. They work well for almost all channels, and are used for system broadcasts as well as for pairs that have not yet adapted to their channel. Both modes are very reliable.

For efficiency, reliability, and the deterministic latency needed by multimedia applications, HomePlug AV uses a beacon-based medium access approach. This also allows coordination among adjacent, interfering networks. Each logical network has a controller that issues a network beacon, which specifies time allocations for specific data streams as well as a period for CSMA/CA access. To handle hidden nodes in a logical network, a proxy coordinator may repeat the beacon.

When a logical network is formed, a Network Membership Key (NMK) is distributed to all its stations. Possession of the NMK defines the stations in the network, whose name is the security level and a hash of the NMK. The controller distributes a periodically changing Network Encryption Key (NEK) to each station, encrypted using the NMK. The NEK in turn encrypts data payloads. The encryption used is 128-bit AES CBC. Transmissions between networks are not encrypted with the NEK.

While communication is very reliable for stations located within a single residence or small office, more than one layer of hidden nodes can cause significant performance problems. Hence, even if confidentiality is not an issue, it is desirable that logical networks be formed, with the controllers exchanging information so that they can avoid interfering with one another.

III. USE CASES

A wide range of node capabilities is anticipated. Some will be computers with a full user interface and a powerful processor. Others will be cheap electronic devices, with perhaps a single button that may be pressed to signal intent. In between, we will have televisions, personal video recorders, DSL routers and the like with various user interfaces and computing capabilities. The protocols have to support devices over this entire range.

When evolving the security specification we considered a number of use (and abuse) cases:

1. Alice lives in a suburban house in the USA, and is not concerned about eavesdroppers. However, her teenage son Bob wants to have a separate logical network for privacy reasons. He wants his network to be able to share a small number of devices with hers, such as the DSL router.
2. Carol also lives in a suburban house, but works from home as a patent attorney. She is aware that private detectives might attempt to compromise her clients' confidentiality. She is not concerned with attacks at a government-agency level of sophistication (she takes no Tempest precautions) but needs at least the equivalent of wire-line security.
3. Dorothy is the private detective trying to break Carol's security. She has hired Eddie, a CS/EE major at the local university, to build an attack tool.
4. Feng lives in an apartment block in Singapore. He is not at all concerned about attacks, but does want his many consumer electronic devices to work. He is highly averse to the embarrassment that would result if one of his gadgets were captured by a neighbor's network, or vice versa.
5. Gordon runs a law firm in a converted warehouse, which is also home to six other businesses. He wants to use power-line communications to provide a small office LAN, and wants to be able to prove if need be that he took appropriate measures to protect his clients' confidentiality.
6. Harry is retired and technophobic. He buys equipment, plugs it in, and it had better work. If not, he will take it back to the shop and demand a refund. He lives in an old semi-detached house that shares a power feed with a neighbor. He suffers occasional power outages and spikes because of poor supply, and also has occasional partial power failures in his house when old wiring or appliances trip one of the earth-return circuit breakers.

These use cases present a range of the scenarios that one may expect to encounter with residential power-line communications.

IV. USABILITY VERSUS CRYPTOGRAPHY

One of the fascinating design questions we faced was the interaction between security and usability. Security engineers tend to think first in terms of establishing a shared key between two devices in order to bootstrap trust. Initiatives such as Trusted Computing may ensure that in the future many devices will come with some form of public-key certificate [9]. The reuse of identities is known to be a hard problem [10]: using names designed for one system in another can lead to a wide range of issues. Certificate revocation is also a problem: in the absence of a dependable update mechanism for many devices, revocation post-manufacture may be hard. But suppose these problems can be overcome (and that we can ignore for now the many cheap devices that are not capable of public key crypto) can we do anything useful with public key mechanisms?

The following example should illustrate the core of the problem. Suppose, for example, that an attacker (Eddie) jams Carol’s TV set-top box using a directional barrage jammer [11], and then attaches a new box of the same make and model to the power line outside her house. Carol suspects a network failure and looks at the network controller app on her PC. It informs her that ‘Set-top box, Brand A, Model XYZ123, cert hash 2E15 3490 AC43 870D 14DA, seeks admission’. If she now assumes that her set-top box somehow got decoupled from her network and presses the ‘admit’ button, she recruits the Trojan – and Eddie is now an authorized user of her network.

If Carol were prudent, she would check the certificate hash against the value printed on the device label – but how many users will do that even once unless they are somehow compelled to do so?

One cause of this problem is that a certificate conveys the authorization of the device manufacturer, while what we actually need is the authorization of the user. Because of the cost of implementing a protocol such as HomePlug (on which we will have more to say later) it is reasonable to assume that almost all attack devices will be adaptations of authorized equipment and would thus come furnished with certificates. So while a certificate can stop a rogue device doing a middleperson attack, this is not our main worry! Whether we are focused on the robustness aspects (as Alice, Feng and Harry are), the privacy aspects (Bob, Carol and Gordon), or the due-diligence issues (Carol and Gordon) the main problem is that a network might recruit a device that it should not.

Assurance of intent

This leads us to a novel view of assurance. Normally, security may be measured according to whether an RSA key is 1024 or 2048 bits long, or whether an operating system has been evaluated to a particular level. But here the key element of assurance is whether the user has assented to a device's joining the network by performing a positive action.

There are some circumstances where high assurance of intent can be conveyed by unambiguous physical actions. For example, in the Resurrecting Duckling protocol, devices are physically touched together to set up initial key material [12]; and technologies such as near-field communications may provide the opportunity to do something similar. However, our protocols are intended for use in a broad range of low-cost devices, many of which will lack extra electrical connectors, near-field capability, or even decent user interfaces. All we can guarantee is that each device has a reset button and a label with a unique high-entropy number.

High-assurance device recruitment, for present purposes, therefore means entering a high-entropy string (such as 2E15 3490 AC43 870D 14DA in our above example) either manually or using a suitable trusted device. Low-assurance recruitment means confirming the identity of the candidate device using simple actions such as pushing a button in response to a flashing light.

Note that we require that the string be entered, rather than just confirmed! Thus, for Carol to enroll Eddie's Trojan set-top box, she would have to obtain this string somehow and enter it into her network controller app. The most likely attack would be some variant of phishing: Eddie would send her an email pretending to be from her satellite-TV provider and asking her to enter the code in order to enable an upgrade to her service. (Controller apps should therefore contain phishing defenses.)

The value added by certificates

Setting up a public-key infrastructure to certify the keys loaded into a large number of consumer electronic devices would be extremely expensive. As noted above, the Trusted Computing Group is working on the problem, but we would not like either to duplicate their effort or delay the launch of HomePlug AV until their system is deployed.

Another possibility is to use public-key crypto but without certificates. In such a scheme, which formed part of our initial design, each capable device generates a public key and sends it to the controller on registration. The controller then uses them to set up temporary encryption keys that in turn protect the network master key. The risk of a man-in-the-middle attack can be dealt with at the high-assurance level by getting users to enter hashes of keys, and at a medium-assurance level using the characteristics of the physical layer (by sending public keys using the low-bitrate assured broadcast mechanism). The more important verification of intent – that the right device is being recruited – would come with high assurance from manual key-hash entry if that option were used, and otherwise at low assurance using confirmation mechanisms to be described below.

This design exercise taught two things. First, there is little benefit gained from public-key certificates. If high assurance of intent is required, and obtained by the user typing in a certificate hash, then the user can as easily type in a key hash directly, and the huge expense of a central PKI can be saved. Second, if the user has to type in a string per device in order to obtain high assurance, then this string might as well be an AES key. That way, we can dispense with the public-key crypto and we no longer have to provide separate mechanisms for cheap devices that cannot do it – with all the attendant complexity of multiple security levels and multiple modes of operation, as well as the increased risk of bugs and blunders.

Device Confirmation

Since no mechanism other than manual key establishment gives sufficiently general high assurance of intent, we decided to use manual keying for high-assurance operation.

But manual key establishment may often be excessive or impractical. A customer using power-line communications to hook up her TV, set-top box and hi-fi will probably not care about the security of the content transmitted over her network; she will take the view that this is all broadcast or published material anyway. She will care about network performance, though, and if a loudspeaker she has just bought starts to play music coming from the apartment above, she will want a simple and direct way to put it right – failing which the loudspeaker will go back to the shop. Keeping device returns low is a significant concern for HomePlug licensees. A significant part of our design exercise therefore focused on the usability of mechanisms for device recruitment, confirmation and revocation.

A user may cause a station already in the network to recruit a new station. If she is operating a network controller with a proper user interface – say, a network controller app on her PC – this is simply a matter of selecting 'enroll a new device' on a menu. If the controller does not have a proper UI, she will press a button that puts it into 'recruit' mode.

The device to be recruited may be configured by the manufacturer to enter 'recruit me' mode by default when it is first powered up, or this may require an action such as pressing a 'recruit me' button. The two devices run a key-establishment protocol (described in the next section) that establishes a temporary encryption key (TEK). This provides the two stations with a reasonably confidential channel. Using this channel, the user can test the new network station, which may be simply by operating it (e.g. trying to play music through a new loudspeaker).

The user will have to reset the station if it is recruited into the wrong network. Every HomePlug compliant device must have some means of resetting the device, including the security state. Thus if you buy a new loudspeaker, plug it in, and hear someone else's music after it is recruited by your neighbor, you will perform some action such as holding down the 'recruit me' button for three seconds in order to reset it. The device will then blacklist the network that it just attempted to join, and will try to join all other reachable networks first before it tries that network again.

When a network recruits the wrong device, it is more problematic. It is anticipated that most users will have a controller with a decent user interface, whether as part of the device itself or exported via a browser (e.g. where the controller is a DSL router). This should allow the user to minimize the chances of recruiting the wrong device inadvertently, but it is not effective against spoofing (deliberate or coincidental, as when two neighbors shop together and purchase the same type of equipment). Regardless of how a wrong device is recruited, the only way to remove the rogue is to reform the network with the desired devices. If a device with a decent user interface is available, then the user may elect to use the high assurance mechanism rather than wrestle with the button-pushing method.

V. Mandatory Security Modes

Following the above analysis, we decided that we needed two modes that must be supported by all implementations, regardless of the capabilities of the device. These are Secure Mode and Simple Connect Mode.

Secure Mode, which involves manual key entry, is very similar to two of the key distribution mechanisms that were supported in HomePlug 1.0, but with one more layer of keys. User experience with HomePlug 1.0 has been very positive, with few returns. Its intended environment, however, is rather different, since it is data-centric and thus is used in networks with at least one capable computer. Users can easily enter passwords into the computer for secure operation. Mechanisms using this will continue to be available in HomePlug AV, though a number of details have been improved over HomePlug 1.0. For example, device passwords must now be 12 alphanumeric characters long rather than eight.

Simple Connect Mode improves over the unprotected mode of HomePlug 1.0, which allows stations to use a single key derived from a fixed password, "HomePlug." While unprotected mode supports a 'plug-and-play' experience for the user, it has the potential to create serious performance problems when the default network becomes large, as noted above. Hence HomePlug AV includes a more sophisticated approach – device authentication that requires minimal user interaction to signal intent, and incurs minimal increased cost per station. The latter consideration is very important with low-end consumer electronic devices, which may not even have a processor apart from the dedicated chip which just implements the basic standard.

Secure Mode

In Secure Mode, key distribution is effected manually. Working at a device with an interface that

permits alphanumeric entry, the user enrolls each other device into its logical network by entering into the controller a Device Password (DPW) that is normally printed on the label stuck to the equipment. The DPW must be at least 12 characters long, giving at least 72 bits of key entropy, and it may be longer. This is hashed to a Device Access Key (DAK), which in turn encrypts the Network Membership Key (NMK). Possession of the NMK enables a device to join a network. The mechanism for creating a key from a password is the PBKDF1 function, as shown in the PKCS #5 v2.0 standard, Password-based Cryptography Standard [13], using truncated SHA-256 as the underlying hash algorithm [14].

The advantage of Secure Mode is simplicity, both of implementation and of operation. Secure Mode is the correct choice for Carol and Gordon in our above use cases, and perhaps for Bob. It has two main disadvantages, especially for the more casual user. The first is that, if wireless LAN products are any guide, many users will not want to make the effort to enter passwords. The second is that it may not be feasible to enter a password for every device – the network might have no device with a keyboard to act as controller, or a device might have no known password (e.g., its label has fallen off or become unreadable).

An alternative in this mode is for the user to choose a network password (NPW) and enter it into each device, where it is hashed to form the NMK. It is possible for the device itself to generate a random NPW and provide it to the user for later use. Manual password entry is discouraged because of the risk of weak password choice, and because most devices will not have interfaces for password entry. However, password entry at network devices provides a compatibility option whereby an NMK can be distributed by other protocols. We will return to this issue later.

To make things more formal, we want Secure Mode to provide the following assurances. First, a network station should not be able to join a logical network unless the user by positive action expresses confidence that it is equipment she wants to add; and stations within a network should enjoy message confidentiality, integrity and authenticity. We assume that all equipment so added to a network by the authorized user is trustworthy and behaves according to the HomePlug specification.

Simple Connect Mode

The objective of Simple Connect Mode is to ensure that casual users can get as close to a ‘plug and play’ experience as is possible while avoiding the risk of creating unmanageably large networks. They should be able to ensure that the devices in their home, and no other devices, are bound to their network, without having to intervene in system configuration or management any more than strictly necessary. If possible, things should just work; else binding a device to a network should involve just a button-push. Even if a recently-purchased device binds to a neighbor's network by mistake, recovery should be easy, and the sequence of steps should be intuitive: something like ‘press the reset button until it works.’

At our first pass at the specification, we started off with an ‘unprotected mode’ in which all devices use the same default NMK (as in HomePlug 1.0). There, users who do not bother with security will have all their devices join a default network, and security will never get in the way. This is ideal for an isolated household with no opponents. It may even be tolerable where occasionally two houses’ networks link up, depending on the applications in use; if Harry's DSL line gets used unwittingly by his neighbor, then maybe no harm is done. However, as applications get complex there will be problems; and regardless of the applications in use, network amalgamation is not acceptable in large shared premises such as apartment blocks. The result is a huge network many of whose stations are not directly accessible to the controller, causing a large drop in efficiency.

Our first pass at a fix for this involved public-key cryptography, which we abandoned once we understood its limitations as discussed in the last section. The current mechanism is much simpler. Each network has one or more user-interface stations that can introduce new stations. A basic UI station has a single ‘admit’ button. On acquiring a new device, the user presses the ‘admit’ button and then plugs in the device to the mains for the first time. On power-up, the new device may seek an open network to join, or the user may press a button on the new device to cause it to search for an open network. The local network remains open for a fixed period of time after the ‘admit’ button is pressed, and so with high probability the device sees only one welcoming controller. (If it sees more than one, it decides based on signal strength.)

Once the device has bound with the controller – which involves operations such as synchronizing with its beacon signal and exchanging tone maps – a key exchange takes place. Each device sends the other a nonce, and the hash of these nonces is then established as a Temporary Encryption Key (TEK). The TEK is used to protect a proper NMK, which is then used as before to protect working keys.

Given that the goal is robust communication rather than security, it would be acceptable for the key exchange to take place entirely in the clear; there are other applications in which initial key establishment is not the critical aspect of protection [15]. However, the characteristics of the HomePlug physical layer allow us to do somewhat better than that, and at zero marginal cost. We note in passing that the use of RF channel characteristics in communications security has a long history, from spread-spectrum and meteor-scatter radio to more modern ideas such as the use of radio channels with fading as a 'wiretap channel' mechanism for key exchange [16].

From the user's point of view, Simple Connect resembles Buffalo Technology's AirStation OneTouch Secure System (AOSS) [17] and BroadCom's Secure Easy Setup (SES) [18,19]. However, these technologies use complex public-key cryptosystems and protocols. Although version 1.0 of the HomePlug AV specification provided for an optional public-key protocol with user confirmation, complexity and cost considerations precluded this option from mandatory inclusion in the specification. Once we had studied the costs and benefits of public-key provision, even optional inclusion in the standard was dropped. We realized that the attack described in section IV above undermines the value of using public key exchange with simple confirmation protocols where the challenge is to tell genuine equipment from genuine but tampered equipment.

Security of Key Exchange

The security analysis of this tone-map key exchange mechanism is interesting. First discussions reveal a serious cultural gap: while a traditional cryptographer will consider attacks on Simple Connect mode communications to be 'obviously' almost trivial, a communications engineer will consider them to be 'obviously' almost impossible.

The cryptographer's viewpoint is that the protocol traffic in the initial key exchange (including both the nonces) is all sent in the clear, and so a capable opponent who observes the exchange can derive the TEK and thus the NMK.

The communications engineer's viewpoint is that the tone-map negotiation uses low-bit-rate broadcast communications – in effect a dependable broadcast channel – so it is difficult to mount a man-in-the-middle attack which would leave the attacker sharing an optimal tone map with each end. As for passive attacks, the key exchange uses high bit-rate communications, which are hard for other stations to decode – even given knowledge of the tone maps – because the signal-to-noise ratio will in general be too poor at different locations for many of the carriers (that is why tone maps have to be negotiated). Furthermore, for an attacker outside the premises, the signal to noise ratios for almost all carriers will be worse than those for a pair of stations inside the premises, at least in one direction. Using the hash of the two nonces requires the attacker to be able to demodulate traffic in both directions. As chips sold by HomePlug and its licensees will not support such attacks, an attacker would have to produce a partial implementation of the HomePlug protocols. This would not only be unlicensed and thus unlawful; it could also be expensive.

A full implementation of the HomePlug protocol might take 30 people 3 years and cost $15m; a very bare partial implementation, just enough to monitor any observable traffic, would likely be a PhD project rather than a summer project. The attacker would have to start with perhaps $100,000 worth of professional test equipment. (Of course, advances in software radios may bring costs down over time, and professional test equipment may end up on the second-hand market.)

Even so, the attacker would have to be smart. Perhaps he can flood the target power-line network with cleverly designed noise that downgrades the tone-maps to relatively low-bitrate communications, and subtract out the noise again to get the nonces. However, he would have to keep on jamming in order to collect the encrypted data traffic; and presumably the target would notice the performance degradation.

Also, to compromise Carol's network (in the attack taxonomy discussed above) two further things would have to happen. First, Carol would have to run in Simple Connect Mode rather than Secure Mode, and second, Dorothy would have to be monitoring Carol's power-line traffic at the very time when Carol was adding a new device to the network. (In theory, Dorothy might give Carol a present of an attractive device that had the label missing, in the hope of

causing a switch to Simple Connect Mode – but Dorothy could just as easily give Carol a device that operated correctly in Secure Mode but was Trojanned in other ways. If you connect untrustworthy kit to your network, then layer 2 defenses cannot buy you much.)

To sum up, a middleperson attack on Simple Connect mode key exchange might just be possible for Eddie, but would cost him a lot of work, and success would not be certain. A private detective prepared to stake out a target residence with a technician and a vanload of surveillance equipment would collect much more through other channels, from phishing scams and laser microphones, through flowers and other presents containing bugs, to Tempest; and if Carol is even potentially facing such an opponent, then she is grossly negligent not to use Secure Mode.

Returning now to Planet Earth, the robustness concerns mostly have to do with failures rather than attacks. For example, what happens if the power fails in half of a customer's house, knocking out the controller? The controller issuing the beacon always maintains a hot backup, to take over if it fails. This does not cause a change in the NMK or even the NEK. Should the old controller return, it will rejoin (using the NMK that it remembers) as any other node would.

To make things more formal, we want Simple Connect Mode to provide the following assurances. It should be hard for another logical network to capture a user's equipment, but easy for him to reclaim it once he realizes it has been captured; it should also be easy for him to expel an alien station captured by accident. It should be easy to identify equipment reliably despite limited user interfaces. The specification must keep complexity, cost and time-to-market reasonable; in particular it must support out-of-the-box, low-return-rate products. It must also be possible to reset a device and sell or give it to someone else.

Switching Security Modes

Having two security levels in a network potentially raises many of the problems associated with multi-level secure systems [20]. For example, a user could end up with two networks at different levels, but since she must have a device with a capable UI in order to have set up a Secure Mode network, we expect that she will have diagnostic software with which she can view the connected devices and their security levels, and thus diagnose the problem. She can then choose to downgrade the Secure network, or upgrade the SC network.

Making downgrading too easy would undermine the value of Secure Mode, so we ensured that an NMK for a Secure network will be different from the NMK for the same network run at Simple Connect. It is up to the vendors of equipment suitable for use as controllers to provide, if they wish, a means of distributing an SC-level NMK using already-established DAKs. This can provide a centrally-managed way to downgrade a network.

We recommend that devices with a single push-button return to SC on reset. Otherwise it might be difficult to get a device from Secure mode to SC – say, if the label had fallen off and the controller that knows its DAK becomes dysfunctional.

Note that the existence of two separate security modes, associated with the NMKs and hence the networks, is a departure from other commercial approaches using a button-push approach, such as SES. In SES, a key that had been previously distributed using more secure methods can be distributed among SES-compliant devices using SES, whereas in HomePlug AV, securely distributed keys must not be distributed using the more vulnerable button-push mechanism. Keeping keys at the ‘Secure’ and ‘Simple Connect’ levels separate from each other permits much greater assurance: Carol and Gordon know that their master keys were never, and will never be, distributed using the button-push method.

VI. Optional Security Modes

Manufacturer keying

The standard also supports an optional security mode in which a manufacturer installs an NMK in equipment sets. For example, someone selling packs that contain a home DSL router and three wireless LAN base stations might install a different, randomly-chosen NMK in each pack, to guarantee plug-and-play performance with no user intervention. However, here there remains an option for the user to enroll the devices in a larger network by either the Secure Mode or Simple Connect Mode mechanisms.

External keying

Trust can also be bootstrapped from other layers or networks. The home of the future is likely to have multiple communications modalities – wireline phone, DSL, Bluetooth, UWB, Near Field, HomePlug and

goodness knows what else. These will interact in various ways. For example, a GSM or DECT mobile phone might act as a home controller, or Near Field Communications might be used to implement a bonding protocol under which the user recruits a device to his network by placing it on top of the TV when he first plugs it in after a reset.

The specification therefore supports key distribution via higher layer protocols, in order to permit use by both existing and yet-undefined key distribution mechanisms. These generally appear to HomePlug devices as though the user had typed in the NPW directly to the device.

Two approaches that have recently been heralded are the USB-stick approach of Windows Connect Now (WCN) proposed by Microsoft or Aladdin, and the Near Field Communication (NFC) proximity approach pioneered by Philips and Sony.

In the WCN approach [21], the user sets up security parameters on a master station, then loads parameters for other stations into files that are transferred to a USB-based removable storage device (flash drive). This flash drive is then inserted into the other devices, which find and read the appropriate security configuration file to set keys and other protection parameters. From a practical use standpoint, this approach requires users to interact with a fairly capable interface device, so they should be able to enter DPWs on it just as easily. Equally significant, the inclusion of a USB port in the bill of materials and in fabrication is likely to raise the cost of including this technology on simple devices (such as speakers) above the acceptable price points.

The WCN approach is supported by the HomePlug standard through direct NMK entry. When the NMK is derived from an NPW, only the NMK is sent across the interface to be loaded on the station, which does not know where the NMK came from. So the NMK may be obtained from a configuration file on a flash drive just as easily as from a hashed NPW entered through a rich user interface.

Aladdin also has USB flash-drive tokens, but these are mostly for user authentication on hosts and networks. The USB devices they make, however, are more than just storage devices, and have smart card capabilities [22]. They could support USB token-based password management in power-line systems. Objections to use of these systems are similar to those for the WCN approach, and, like the WCN approach, they can be supported at the host level if desired.

NFC standards have been spearheaded by Philips [23] and Sony [24], and standards are now set by ECMA and ISO [25]. Similar in some respects to Radio Frequency ID tags (RFIDs), NFC operates in the 13.56 MHz band. However, unlike RFIDs, NFC allows interactive data exchange at a distance of 10-20 cm., rather than simply remote read of a fixed value. When two NFC-compliant devices are brought close together, they detect each other; they negotiate what data they can transfer and how they can do it. For authentication, this may allow a “wand” to be used to transfer keys to all suitably compliant devices.

While this could support a very desirable user experience, again the cost for inclusion of this technology in inexpensive consumer electronics products may be too high for many manufacturers. Also, the utility of such approaches diminishes as they become less ubiquitous. Still, as with the USB-based approach, NFC authentication is supported by the baseline HomePlug protocol – it can be implemented as the host device downloading the NMK directly to the station.

A possible future concern is that NFC may also be used for reading an RFID attached to the device. This could contain the device’s DPW, which could then be used to derive the DAK and provision the NMK using the DAK-based approach, as though the user entered the DPW by hand. While this approach is attractive from the perspective of cutting the per-device costs, it raises serious concerns over the degree to which the DAK is protected. Given recent results in reading RFIDs from much greater distances than advertised, use of RFIDs in this manner could open a large hole in the security of the system (even the RFID Journal admits that passive RFIDs can be read up to 20 feet away [26]).

The standard is agnostic about how a DAK is acquired; the network does not know whether the DAK was derived from a DPW that was entered manually, or from some kind of automated reader that scanned the device for its DAK. A vendor who implemented DAK scanning would have to consider further issues, such as whether Eddie could set up an attack in which his equipment broadcast a DPW and waited for Carol’s controller to read it.

In general, OEMs designing key-management protocols that use multiple communications modes need to beware of a wide range of security engineering issues, from naming problems through API defects to protocol interactions, compositional issues, policy incompatibilities and attacks based on

changing environmental assumptions [20]. Connecting two secure systems together is almost always harder than it looks.

Fillgun

Going back once more to pre-public-key technology, one option is the fillgun. These were devices used to load key material into military cipher equipment. The power-line equivalent might be sold as an adapter, with a male plug and a female socket. The user plugs it into the wall, then plugs each appliance into it in turn, pressing the appliance reset button as he does so. The fillgun loads an NMK into each of them: a simple solution for the consumer who wants security but can't be bothered to type DPWs into his TV, and perhaps also for the small business that's seriously worried about phishing. Physical contact was the traditional method of keying cryptographic devices; its simplicity and usability have led to a resurgence of interest [12].

A fillgun could also use the existing Simple Connect mode. The device can have a low-pass filter between the female socket and the male extension cord, and between the filter and the female socket is a HomePlug AV chip with an embedded host. The embedded host has a primitive user interface that allows a new NMK to be generated when requested by the user. This node always behaves as a controller, and is always willing to distribute the NMK that it has to a new station (i.e., any device that is plugged into its female socket). The usual Simple Connect Mode protocol works the same as before, only now there is no possibility that an eavesdropper can demodulate the key exchange messages, as the low-pass filter eliminates the signal containing these messages. This approach has the decided advantage that neither the device's DAK nor the NMK can be read remotely (as in RFIDs and potentially, NFC), and there is no additional cost per device – only the cost of the fillgun itself.

In fact, a nervous user could even employ devices already present and in use in the home to get an extra level of protection. Many surge protectors are also effective low pass filters. Hence, if a user just plugs a controller into the same surge protector as a new device that is to be recruited to the network, then presses buttons on both, the Simple Connect key exchange mechanism may become significantly harder to attack.

Resurrecting Duckling

This protocol [12] enables manufacturers to make products theft-resistant by ensuring that a device once bonded to a controller cannot be properly reset without the cooperation of that controller. This can be easily implemented on top of HomePlug. Although we recommend that manufacturers return a device to simple-connect mode by default when the reset button is pressed, this is not mandatory; devices may be manufactured (or configured later) to reset to Secure Mode. The binding between such a device and its controller can be made permanent by removing the label. A thief who steals the device will not know the DPW, and thus will be unable to introduce it to a network, short of reverse-engineering it.

Of course, with many low-cost products, a default of theft-resistance would likely annoy the legitimate owners much more than any burglars. However the theft-resistance facility of the Resurrecting Duckling protocol is available when needed.

VII. CONCLUSIONS

We have discussed some interesting trust problems with home networking, and described how they are tackled in the next generation of power-line communications. The main problem is that users may recruit the wrong devices to their networks, and conventional trust mechanisms such as public-key certificates simply don't deal with this. To check that you're recruiting the right device you need to check its label, or perform some other physical action with it; and in that case, there are cheaper ways to do things.

In our design, we provide two simple modes of operation: Simple Connect Mode (which prevents accidental recruitment) and Secure Mode (which blocks more sophisticated malice). These correspond to low and high grades of assurance about user intent – an issue to which we believe insufficient attention has been paid so far. We also provide the hooks necessary for licensees and third-party vendors to create their own approaches, and to support competition between different network personalization technologies.

Acknowledgements

We are grateful to Frank Stajano and to the anonymous referees for comments that improved this paper, and to colleagues in the HomePlug project for feedback at various stages during the design process.

REFERENCES

[1] Brown, P.A., "Power line communications – past, present, and future", Proceedings of International Symposium on Power-line Communications and its Applications, Sept 1999, pp. 1--8

[2] Lee, M. K., R. Newman, H. A. Latchman, S. Katar, and L. Yonge, "HomePlug 1.0 Powerline Communication LANs – Protocol Description and Comparative Performance Results", International Journal on Communication Systems on Powerline Communications, May 2003, pp 447–473

[3] HomePlug Powerline Alliance, "HomePlug AV 1.0 Specification," December 16, 2005 (visit http://www.homeplug.org)

[4] HomePlug Powerline Alliance, "HomePlug AV White Paper," August 18, 2005 (last read May 25, 2006, at http://www.homeplug.org/en/docs/HPAV-White-Paper_050818.pdf)

[5] Afkhamie, K. H., S. Katar, L. Yonge, and R. Newman, "An Overview of the upcoming HomePlug AV Standard," proceedings of International Symposium on Powerline Communications (ISPLC 2005), Vancouver, BC, 2005, pp. 400-404..

[6] Katar, S., R. Newman, H. Latchman, and L. Yonge, `Efficient Framing and ARQ for High-Speed PLC Systems', proceedings of International Symposium on Powerline Communications (ISPLC 2005), Vancouver, BC, 2005, pp. 27-31.

[7] W. David Gardner, "Wireless Survey: Many Nets Open To Security Breaches", Information Week, Mar 10, 2005 , see http://www.informationweek.com/story/showArticle.jhtml?articleID=159400875.

[8] Prasad, R., van New, R., '*OFDM Wireless Multimedia Communications*', Artech House, Norwood, MA, 2000.

[9] *X.509, The Directory – Authentication Framework.*, CCITT, ITU-T, 1988; the IETF version is available as '*Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*' at http://www.ietf.org/rfc/rfc3280.txt

[10] R. Needham. "Names" In S. Mullender, (ed.), *Distributed Systems*, Addison-Wesley, 1993, pp. 315—327.

[11] D Richardson, '*Techniques and Equipment of Electronic Warfare*', Salamander Books, ISBN 0-8601-265-8

[12] Frank Stajano, Ross Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks", Security Protocols, 7th International Workshop Proceedings, 1999, 172—194.

[13] RSA Labs, PKCS #5 v2.0 standard, Password-based Cryptography Standard.

[14] FIPS 180-2, NIST, "Secure Hash Standard," August 26, 2002, (including the change notice dated February 25, 2004, concerning truncation)

[15] R Anderson, HW Chan, A Perrig, "Smart Trust for Smart Dust", ICNP, Berlin, Oct. 5–8 2004, pp 206–215

[16] J Barros, MRD Rodrigues, "Secrecy Capacity of Wireless Channels", IEEE Symposium on Information Theory 2006

[17] Buffalo Technology, "AirStation OneTouch Secure System (AOSS)," white paper, Oct. 2004, (last read May 24, 2006 at http://www.buffalotech.com/documents/pdf/AOSS_WP_Final.pdf)

[18] Broadcom, Securing Home Wi-Fi Networks: A Simple Solution Can Save Your Identity," white paper Wireless-WP200-x, May 21, 2005, (last read May 25, 2006, at http://www.54g.org/pdf/Wireless-WP200-RDS.pdf)

[19] Moran, Joseph, "Push-Button Wireless Security," Small Business Computing.com Web Management ezine, December 2, 2005 (last read May 24, 2006 at http://www.smallbusinesscomputing.com/webmaster/article.php/3567981)

[20] R Anderson, '*Security Engineering – A Guide to Building Dependable Distributed Systems*', Wiley 2001

[21] Bowman, Barb, "Set up a secure wireless network using Windows Connect Now," Microsoft XP ezine, June 13, 2005, (last read May 25, 2006, at http://www.microsoft.com/windowsxp/using/networking/learnmore/bowman_05june13.mspx)

[22] Alladin, "Make Your Token Authentication Solution a Reality with a Token Management System," white paper WP_eToken_TMS, March 1, 2006, (last read May 25, 2006, at ftp://ftp.aladdin.com/pub/marketing/eToken/White_Papers/WP_eToken_TMS.pdf)

[23] Harold, Peter, "Close up and in the Comfort Zone," Philips Password, issue 24, Sept. 2005, (last read May 25, 2006, at http://www.research.philips.com/password/archive/24/downloads/password24.pdf)

[24] Sony, Felica product site, (May 25, 2006) http://www.sony.net/Products/felica/index.html

[25] ISO, ISO/IEC 21092 Standard – Near Field Communication -- Interface and Protocol (NFCIP-1) http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=38578&ICS1=35&ICS2=100&ICS3=10 or download at http://isotc.iso.org/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm

[26] RFID Journal FAQ, "Privacy and Data Collection," http://www.rfidjournal.com/faq/28/138