Chapter 17

Deploying and Configuring Intel® Active Management Technology

There is no stigma attached to recognizing a bad decision in time to install a better one.

—Laurence J. Peter (1919–1988)

Intel® Active Management Technology (Intel AMT) is an interactive subsystem that responds to various kinds of inputs. Most hardware or software products need some kind of initialization or setup before becoming operational. Intel Active Management Technology is no different in this regard. The IT administrator of the Intel AMT computers needs to properly configure Intel AMT according to his IT environment so that he derives maximum benefit from deploying this technology. Intel AMT offers a wide choice of options and tools to set up and configure it, catering to various usage scenarios and needs. The goal of this chapter is to help you understand the various scenarios that are supported for configuring Intel AMT, the technical details of the options that are available for configuration, and the tools that Intel offers to do the configuration. This chapter does not provide a step-by-step guide to configuring Intel AMT, nor does it provide exhaustive lists of configuration parameters or settings and their descriptions. Please refer to Intel AMT manuals for such information. Instead this chapter dives into the design aspects of Intel AMT setup and configuration.

What Is Setup and Configuration for Intel® AMT?

Intel AMT requires some information about the network and IT environment in which it needs to operate. This includes parameters like network domain name, IP address, DNS information, 802.1x information and credentials, Wi-Fi† settings, Active Directory credentials, administrator passwords, permissions, and so on. Intel AMT also needs to be initialized with various policies and settings that control how Intel AMT behaves, such as System Defense policies, power policies, and Agent Presence policies.

The IT administrator of Intel AMT needs to configure these settings, policies and parameters. Only then will Intel AMT be able to operate correctly and as expected. For example, let us consider the scenario of a wireless router that many of you might have configured in your homes. A wireless router is similar to Intel AMT in some respects such as both are embedded devices, both are connected to wired/wireless networks, both need to be managed (remotely, in most cases) by an administrator of some sort, and both provide services to some end users. Any wireless router provides various configuration settings that you (as an administrator of the router device) can set according to your needs, such as setting an administrative password, setting up an IP address, DHCP and an IP address pool, enabling/disabling security settings (such as WEP or WPA), port filtering/firewall configuration, and so on. Once this is done, your router operates based on your settings and ensures that your home network works the way you wanted. Analogously, Intel AMT needs to be configured for it to work in your networking environment as you expect. However, the scenarios in which Intel AMT operates are obviously very different from wireless routers, so it would be a mistake to take this analogy too far. The following section describes the various scenarios in which Intel AMT may need to be configured. Subsequent sections describe how Intel AMT gets configured in each of these scenarios.

Deployment Scenarios

Intel AMT is a product that can be used by businesses of any size—Fortune 500 enterprises, large businesses, or small and medium businesses. Our consideration of a small business is one that has less than 100 employees; a medium business is one that has 100 to 1000 employees; and anything over 1000 employees is a large business. Each of these classes of businesses may have a different set of requirements for how they would like to configure Intel AMT. In order to have a broad and general deployment capability, Intel AMT needs to work well in each of these deployment scenarios. Following are just some of the examples of deployment scenarios for Intel AMT.

- A large enterprise may have a strong need to do a fully automated configuration of Intel AMT, but is willing to bear some costs associated with it.
- A small enterprise may be extremely cost conscious, but may be willing to compensate for this by doing a few manual steps.
- A large business may have several branch locations but a centralized IT administration organization. Such an organization may prefer to opt for a fully remote configuration of Intel AMT because it may be too expensive to send a technician to each computer to configure Intel AMT. Such an organization may already have or be willing to deploy some central servers and tools to achieve remote configuration.
- A one-person IT department of a small business (all of it located in one building) may find it totally acceptable to walk up to every computer in the building to configure and turn on Intel AMT, perhaps on a weekend.

Factors to Consider

The examples above demonstrate that several factors could influence the option that an IT department chooses to configure Intel AMT. These could be the cost of resources (tools, additional hardware/software), geographical distribution, the number of machines that need to be configured, the security and trust the IT department places in the infrastructure used for configuration, and so on. These factors govern various configuration options that businesses choose, such as who configures Intel AMT, where is it configured, and when is it configured. Let's look at some of these below.

Who Can Configure Intel® AMT?

Of course, the main control or authority in configuration of Intel AMT resides with the IT administrator of the computer. But the IT administrator may have the original equipment manufacturer (OEM) configure Intel AMT with some initial parameters that are specific to his enterprise, such as the domain name of the enterprise. The OEM may offer such customization options for a group of computers being shipped to that organization. This would require the customer to specify the customization details to the OEM at the time of placing the order. These customized parameters would be stored on the flash device in the computers. If the OEM does not do any customization of configuration parameters, then the IT administrator may do some initial configuration personally by turning on each computer and doing some manual operations on Intel AMT (via the BIOS screens). It is also possible that Intel AMT gets configured in a fully automated way without any particular configuration being done on it beforehand. Lastly, the end user of the computer could also configure Intel AMT in a plain vanilla manner (with reduced capabilities and functionality), without using any special tools or resources. The IT department may weigh several considerations before asking the OEM to configure Intel AMT parameters, such as additional administrative or operational costs involved, or the willingness of the IT department to share any information about its infrastructure with the OEM.

Where Can Intel® AMT Be Configured?

As described above, some initial customer-specific custom configuration of Intel AMT can occur at the OEM's factory, at the time the computer is being manufactured. Alternatively, the organization can buy off-the-shelf computers from the OEM and do the configuration at their own end. Within the organization, the IT departments may do some or the entire Intel AMT configuration themselves before handing over the computers to employees (end users). Or the computers may be shipped directly to the employees, without getting routed to the IT department. In such case, the configuration of Intel AMT would happen at the employee's location, albeit under IT controlled processes and over the intranet of the organization. This is called *remote configuration*. This scenario applies, for example, to organizations having several branch locations without dedicated IT staff being present at each location. Conversely, a well staffed IT department may choose to configure Intel AMT at a specific IT location (such as an IT depot) so that any manual configuration operations (as opted by the IT administrators) can be performed by the IT personnel and not depend on the end users to do the same. This is called *one-touch configuration*.

When Can Intel® AMT Be Configured?

It is possible to configure Intel AMT on a bare-metal computer (a computer that has no operating system installed on it). Such configuration occurs solely over the out-of-band (OOB) network channel. This scenario may occur when an IT department does Intel AMT configuration before loading an operating system image on the computer, soon after receiving the computer shipments from the OEM. This is called *bare-metal configuration*. On the other hand, IT departments may configure Intel AMT some time after they have rolled out the computers to their employees (for example, several months later). In such a scenario, the configuration of Intel AMT would require some assistance from the host operating system to trigger the configuration process, albeit still via the IT controlled processes. This is called *delayed configuration*. Delayed configuration may happen if an IT department is not yet ready to turn on Intel AMT (for various business or technical reasons, such as when the IT department is still conducting pilot tests, or waiting to hire a team that will turn on Intel AMT), but would like its employees to start using the new computers soon after they arrive.

Many possible scenarios need to be covered for successful deployment of Intel AMT. The next few sections describe the various configuration methods that are available in Intel AMT, and how these methods can be effectively used to cater to all of the aforementioned scenarios.

Intel® AMT Setup and Configuration Overview

Intel AMT configuration is done using a configuration server (Enterprise Configuration), or using a browser over a web-based interface. The web-based interface is usually used for pilot projects or small businesses. Large sized (even medium sized) enterprises will usually prefer to use the configuration server mechanism. The configuration server is available as part of ISV management suites, such as those available from Microsoft (Microsoft† System Center Configuration Manager), Symantec (Symantec† Notification Server), and LANDesk† (LANDesk Management Suite). The configuration server establishes a secure connection with Intel AMT and then downloads the configuration data into Intel AMT. The protocols for setting up the secure connection are described in subsequent sections.

Intel® AMT Web-based Configuration

Intel AMT is designed to provide a web-based configuration method for small businesses or for pilot projects in enterprises that want to try out the capabilities of Intel AMT. The beauty of the web-based configuration method is that it does not depend on any third-party software. However, since the configuration process requires manual operation on each individual Intel AMT computer, it is not scalable beyond a few machines (otherwise an IT technician would have to spend several sleepless nights performing the same operations over and over again on different computers).

In the web-based configuration method, the initial configuration of Intel AMT is performed through a specialized BIOS module, available on Intel AMT systems, called the Intel Manageability Engine BIOS Extensions (Intel MEBX). An administrator brings up the Intel MEBX screen by going into the computer's BIOS, and configures various settings in Intel MEBX such as a password, and some network settings. Sample Intel MEBX screens are shown in Figure 17.1 and 17.2. Using the network settings configured in Intel MEBX, the administrator now connects to the Intel AMT subsystem over the network using a browser, and configures it from the browser interface (much like you would configure your home networking router by connecting your browser to the router).

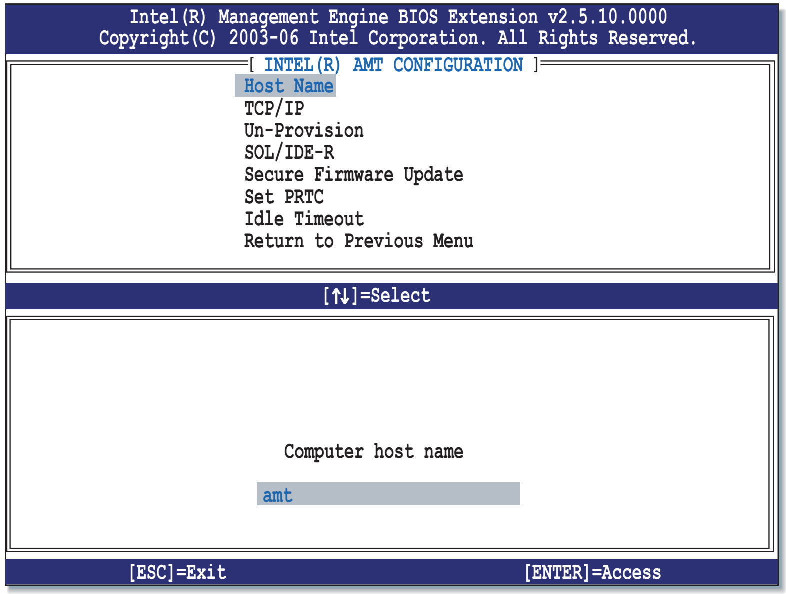

Figure 17.1 Intel® MEBX Screen for Setting Up the Host Name for Web-based Configuration

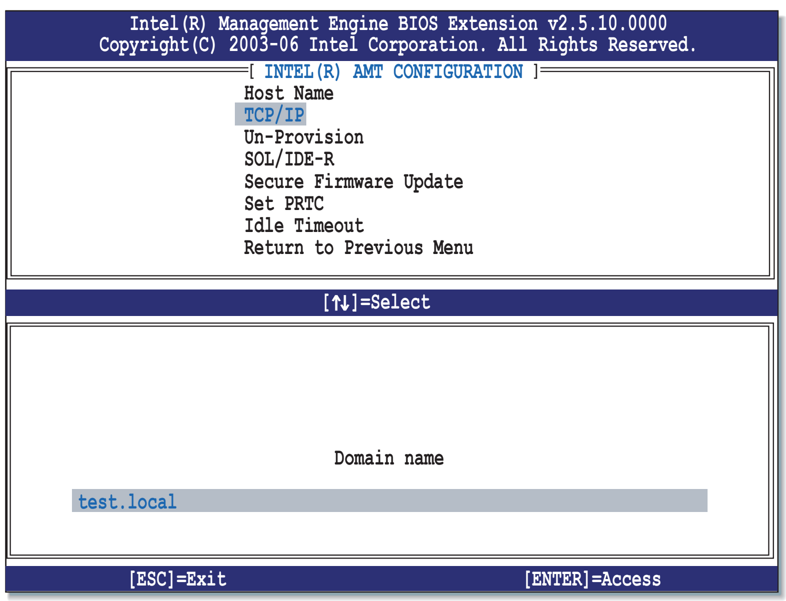

Figure 17.2 Intel® MEBX Screen for Setting Up the Domain for Web-based Configuration

Figure 17.3 shows a sample of the Intel AMT Web page for configuring the various settings in Intel AMT. This Web page is obtained by a web browser after it successfully connected to the Intel AMT subsystem. Note that the default port number used by the Intel AMT subsystem for this web-based configuration interface is 16992.

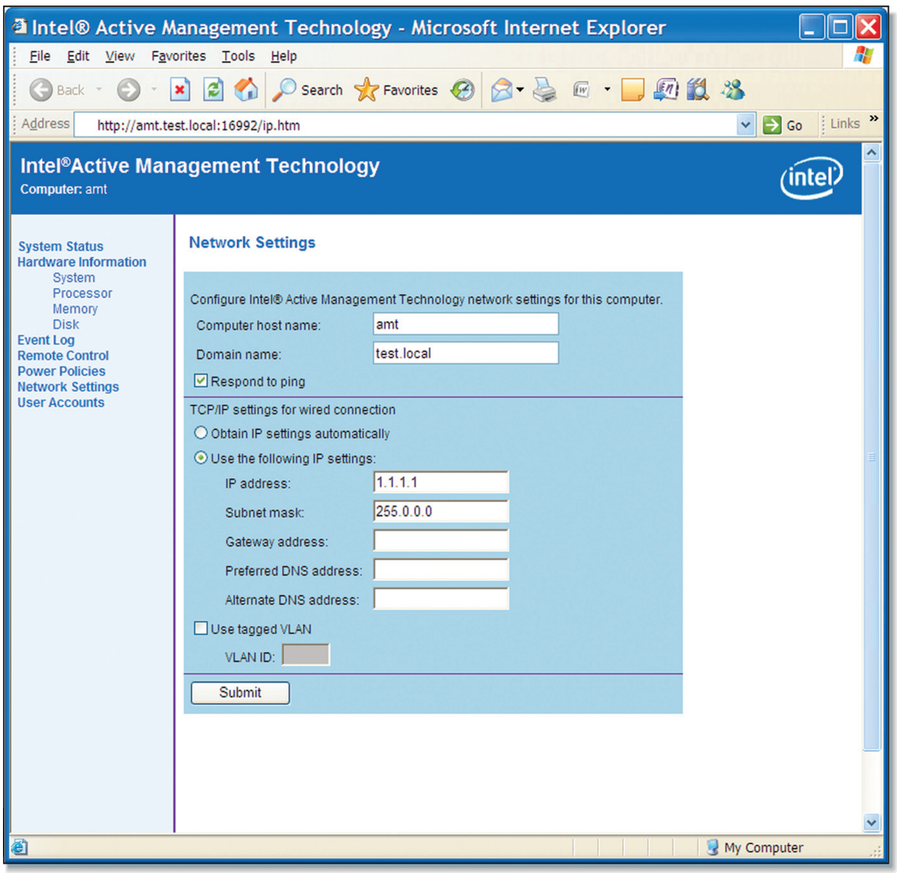

Figure 17.3 Intel® AMT Web-based Configuration

A step-by-step procedure to configure Intel AMT using this mechanism is given in the Appendix A.

Intel® AMT Enterprise Configuration

Configuration of Intel AMT in the enterprise environment is fundamentally based on two protocols, the pre-shared key (symmetric key) based TLS protocol and the asymmetric key based TLS protocol. Certain attributes and properties of these protocols can be adjusted to achieve varying levels of security and configurability. We will cover all these methods in detail in the sections below.

However, before we go into the details of the aforementioned protocols, it is important to note that Intel AMT does not support pure host-based configuration. That is, a piece of software running on the operating system of the Intel AMT computer cannot alone set up and configure Intel AMT without any other authentication or security checks. Even though host based configuration may appear to be a very simple and straightforward approach to configure Intel AMT, it comes with its own risks and issues. If this configuration mechanism were allowed, a virus or malware in the operating system could configure Intel AMT too, causing adverse security consequences for the enterprise. So, as we will see in the mechanisms described below, Intel AMT requires more trust establishment to be able to proceed with the configuration rather than just the mere presence of a piece of software in the operating system environment of the Intel AMT computer. Before the advent of Intel AMT, manageability technologies such as the Alert Specification Format (ASF) and Wired for Management (WfM) were configured through the host operating system. For example, configuration of Intel's ASF management controller was performed through a Windows Management Instrumentation (WMI) provider on the locally installed Microsoft Windows OS. This could be used by any software application to configure ASF, specify ASF policies, and designate remote management servers. From a security perspective, even the malware applications located on the host operating system could exploit the capabilities provided by ASF. However, since those capabilities were limited to alerts and remote power-up/power-down operations, the consequences of such misuse were typically low, as was the overall vulnerability of the system. Intel AMT offers much stronger protections to the enterprise IT departments, including the capability to boot systems from a remotely-situated media and to share data between local and remote software agents. With this in mind and with the ongoing rise in malware vulnerability incidents, Intel AMT requires a more resilient configuration method that can protect against such malware.

Pre-Shared Key TLS based Configuration Protocol

The pre-shared key (PSK) based TLS protocol is based on the TLS-PSK RFC (RFC #4279). This RFC specifies a mechanism by which two parties can establish a secure channel of communication between themselves. See the gray side box for a quick overview of the TLS-PSK protocol.

TLS-PSK Protocol Overview

The TLS-PSK protocol is a mechanism for two interested parties to set up a secure channel of communication between them. The two parties start out by sharing a secret. They then use this secret to derive a TLS pre-master secret. This pre-master secret is then used to encrypt all the data between the two parties. There are several cryptographic algorithms that can be used to encrypt the traffic such as AES, RC4, and so on.

Some of the popular TLS-PSK cipher suites are as follows:

- TLS_PSK_WITH_RC4_128_SHA
- TLS_PSK_WITH_AES_128_CBC_SHA
- TLS_PSK_WITH_AES_256_CBC_SHA

As long as no one else besides the two parties involved on either side of the connection knows the shared secret, they cannot compute the pre-master secret, and hence cannot decrypt the communication.

The TLS-PSK protocol is essentially a modification of the asymmetric key based TLS protocol. In TLS PSK, several of the messages are not required. The overview of the protocol is depicted in Figure 17.4, with respect to the asymmetric key based TLS protocol. The messages that are struck through show the messages from the asymmetric protocol not required in TLS PSK.

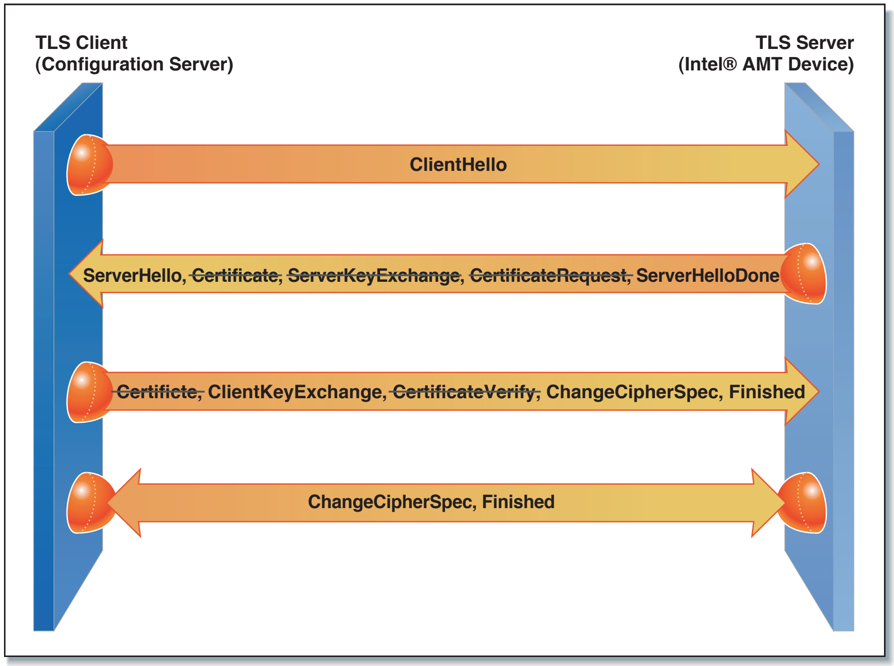

Figure 17.4 TLS-PSK Protocol Flow

In the case of Intel AMT, the two parties interested in setting up a secured communication channel are the Intel AMT subsystem and the Intel AMT Configuration Server (also simply referred to as the configuration server). Intel AMT acts as the TLS server, and the configuration server acts as the TLS client. The starting assumption of the TLS-PSK protocol is that both parties must already share a secret. In our context, we call this shared secret a *provisioning passphrase* (PPS). There is also an associated identifier with each PPS called the Provisioning ID (PID). How the configuration server and Intel AMT get to agree upon this shared secret is the subject of the rest of this section. After the secret sharing has been accomplished, the configuration server and Intel AMT set up a TLS-PSK based secure session between them. Per the TLS-PSK RFC, Intel AMT sends the PID as the "PSK_Identity_hint" value within the TLS handshake, allowing the configuration server to locate the matching PPS value and use it in the session establishment of the TLS session. Once the TLS session has been established, the configuration server proceeds with the configuration of the Intel AMT device with the enterprise-specific information. This is described in the section on configuring enterprise data, later in this chapter.

The PID and PPS

The PID and PPS are strings of characters comprising capital letters A–Z and numbers 0–9. The PID is 8 characters long and PPS is 32 characters long. The reason for restricting the PID and PPS to capital letters was to reduce manual errors in entering the values. By including small letters, the entropy of the PPS could have been increased by a huge amount, or possibly the length of the PPS could have been reduced. But we decided to trade off these benefits in favor of avoiding errors during manual entry. In addition to this, every fourth character of the PID and PPS is a check-character, that is, it indicates whether the previous three characters were typed correctly or not.

```
PID: KRM8-6D6Y

PPS: G6HH-VD5R-DFP2-INJ9-XV6C-KA6E-J27Z-2V15
```

This enables any software that uses a PID and PPS to immediately detect whether these values are properly formed or not.

A Note on the Security Strength of PPS

The PPS is a 32-character value, with each charater having 36 possible values (A–Z and 0–9). But out of the 32 characters, only 24 characters comprise the true secret value. The remaining 8 characters are derived characters (remember every fourth character is a check character). Therefore the total number of uniquely possible provisioning passphrases are 36 ^24. This translates to approximately 125 bits of security.

PID/PPS Sharing Techniques

Intel AMT provides several ways to achieve the sharing of PID and PPS between the configuration server and the Intel AMT susbsystem, as described below.

Manual Entry of the PID/PPS Pair

The PID/PPS pair can be manually entered into the Intel AMT subsystem via a BIOS interface screen of the Intel Management Engine (Intel ME) called the Intel Management Engine BIOS Extension (Intel MEBX). The same PID/PPS pair is entered into the configuration server. The configuration server is expected to expose interfaces to allow entering the PID/PPS. For example this could be a manual interface like a GUI dialog, or a programmatic/scripting interface for automated entry. Alternately, the configuration server could be designed to generate a list of PID/PPS pairs and allow the IT technician to print the list. The IT technician would then go from machine to machine entering a PID/PPS entering a PID/PPS value from this list one by one via the Intel MEBX interface on each Intel AMT machine.

Automated Entry of PID/PPS into Intel® AMT

As you can imagine, manual entry of PID/PPS into Intel AMT machines via the Intel MEBX interface is not a very scalable over more than a small number of machines. To overcome this problem the Intel MEBX has a special capability to detect if a USB thumb drive is plugged into the computer and can read PID/PPS pairs from the USB thumb drive.

The first step in this method is for the IT technician to go to the configuration server and select the option of generating data and loading it on a USB thumb drive. The configuration server generates a list of PID/PPS pairs, formats the records in the appropriate format that would be readable by Intel MEBX, and loads them onto the USB thumb drive that was inserted into the configuration server machine. Next, the technician goes to each Intel AMT machine, plugs the USB thumb drive into it, and powers it on. As soon as the machine boots and the Intel MEBX logic executes, it detects that a USB thumb drive is available. The Intel MEBX logic also detects that the USB thumb drive has a list of PID/PPS pairs loaded in it in the expected format. The Intel MEBX then reads the first available PID/PPS pair and stores that information into the Intel AMT nonvolatile flash memory. It also marks that PID/PPS pair as “unavailable.” Then the Intel MEBX deletes the PPS value from the USB thumb drive. This is done so that if the USB thumb drive is lost and falls into the wrong hands, then the attacker does not get access to the PPS values already configured into Intel AMT machines. After working for a few seconds, the computer will automatically reboot or prompt for reboot. This automated method offers increased convenience and scalability in the act of configuring the PID/PPS pair relative to the manual method.

However, some very large or complex IT departments may find even this mechanism inconvenient to use. There could be several reasons for this. Some of them may be as follows:

- The enterprise is so large, complex, and scattered that it is impractical for a number of IT technicians to go to every Intel AMT machine, plug in a USB thumb drive, power it on and then power it off.
- For some enterprises, the cost of staffing this group of IT technicians may itself be a prohibitive option.
- An enterprise has the practice of shipping newly purchased computers directly to end-user locations without even opening the cardboard packaging. The end user is expected to open the box, take out the computer, plug it into the power socket, plug in the network cable, and start using it. This leaves no opportunity with the IT department to touch the computer before it reaches the end user.

To overcome this obstacle there is yet another way to share the PID/PPS pair that involves the OEM (such as Dell, HP, and so on). However, in reality, this manufacturer could be a supplier to the OEM, or a supplier to the supplier to the OEM. The process of computer manufacturing is fairly complex and involves multiple parties that form a manufacturing chain. To keep things simple, we will group all of these parties into a single category that we call the OEM.

Configuration of PID/PPS by the OEM

The OEM can configure a PID/PPS pair into an Intel AMT computer at the time of manufacturing it, and then ship this computer to the customer along with the PID/PPS using some transmission mechanism such as sending an e-mail message to the customer, printing the PID/PPS pair on a paper and enclosing it in the computer packaging, or storing it on a CD-ROM and packaging the media along with the computer. The customer or the IT technician that receives this computer enters this PID/PPS pair into the configuration server. The OEM uses special low-level software tools that execute on the computer manufacturing line to configure the PID/PPS into the Intel AMT computer.

The above description is meant to explain the technical theory of operation. Of course, it may not be practical for an OEM to configure a PID/PPS for a customer buying a single computer. This method makes more sense when a large enterprise customer orders a fairly substantial number (say, a few hundred or more) machines from an OEM, and negotiates with the OEM to configure the PID/PPS pairs on the computers on the OEM manufacturing line. This saves the effort on part of the customer to configure the PID/PPS into the machines and instead transfers that responsibility to the OEM. We will not go into the economics of this negotiation. Some OEMs may charge the customer for configuring PID/PPS pairs into the machines, while some other OEMs may work in the cost into the overall price without charging an additional amount for this. Customers should discuss these options with their OEMs. For a large order, the OEM may be willing to e-mail a list of the configured PID/PPS pairs to the customer, or send a CD-ROM to the customer.

It is more secure if every machine is configured with a unique PID/PPS pair, however in some scenarios the customer and OEM may negotiate that all machines for that customer's order would be configured with the same PID/PPS pair. This method reduces the number of PID/PPS pairs to be managed and hence may be seen as a case of increased convenience obtained by sacrificing the security aspect. Security is significantly lowered in this case because if an attacker breaks into the hardware of one machine in that lot of machines, and if he knows that all other machines have the same PID/PPS pair, then the security of all the machines is compromised. The benefit to the OEM is that it does not have to configure a unique PID/PPS on each machine. This makes the manufacturing process simpler. The OEM could create a common configuration image for the whole lot, and configure all machines in that lot with the same image without doing anything different from machine to machine. The transmission of the PID/PPS also becomes simpler since only a single pair needs to be sent to the customer instead of a long list.

In all of the OEM configuration mechanisms mentioned above (whether unique PID/PPS pairs are programmed, or a common pair is programmed across several machines), it is important to note that the communication of the PID/PPS pair(s) to the customer must happen in a secure manner. For example, sending the pair(s) over unencrypted e-mail is not advisable. Some better alternatives could be encrypted e-mail, sending a courier/mail with the list in a disk or CD-ROM, and so on. The customer and the OEM should negotiate a secure mechanism that is acceptable for them.

The previous sections described how the pre-shared (symmetric) key based configuration happens for Intel AMT machines. The next few sections describe public (asymmetric) key based configuration mechanisms. In asymmetric key based mechanisms we will see how remote configuration is achieved and yet maintaining protections from attacks.

Asymmetric Key TLS based Configuration Protocol

The asymmetric key based configuration protocol for Intel AMT is the protocol to achieve Remote Configuration. It is based on the TLS standard (RFC #4346), using mutual authentication. This TLS standard specifies a protocol by which two parties can set up a secure channel of communication between themselves using RSA key pairs established by each of them beforehand. There is no need to pre-share any secret such as PID/PPS pairs between the two parties. This is the biggest advantage of this mechanism—no mechanisms need be devised to share secrets, as was the case in the previous protocol, which also was its biggest hurdle. See the gray box below for a quick overview of the RSA key pair based TLS protocol.

The asymmetric key based configuration protocol for Intel AMT is available in several flavors that differ very slightly from each other. We will cover the most general flavor first to get a good grounding into its behavior. Then we will cover the differences across various flavors.

TLS Protocol Overview

The TLS protocol is a mechanism for two interested parties to set up a secure channel of communication between them. Unlike TLS-PSK, the two parties do not have to share a secret. Each party (at least one of them) possesses a private key and a public key certificate. The private key and public key share a mathematical relationship between themselves. The public key of each party is known to the other party. One of the parties uses the other party's public key to send a shared secret. In this manner the two parties end up negotiating a pre-master secret. This pre-master secret is then used to encrypt all the data between the two parties. Several cryptographic algorithms can be used to encrypt the traffic such as AES and RC4.

Some of the popular TLS cipher suites are as follows:

- TLS_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA

As long as no one else other than the individual holding the private key knows that private key, they cannot compute the pre-master secret, and hence cannot decrypt the communication.

The overview of the protocol is depicted in Figure 14.5.

As you know by now, the two parties interested in setting up a secure channel in our case are the Intel AMT machine and the configuration server. Intel AMT acts a TLS server and the configuration server acts as the TLS client. Since we will depend on mutually authenticated TLS, both parties require a RSA private key and public key certificate. On one hand, Intel AMT generates a 2048-bit modulus based RSA key pair early during its lifetime (when the machine was first boots and the Intel ME begins execution), creates a self-signed X.509v3 certificate for this key pair, and stores the certificate and private key in the nonvolatile flash memory associated with Intel AMT. On the other hand, the configuration server uses standard mechanisms to generate a 2048-bit modulus based RSA key pair and obtain a certificate for the public key from a partner commercial Certification Authority (CA). Intel has partnered with some of the leading CAs to streamline the process for obtaining certificates for Intel AMT configuration. The partner CAs at the time of this writing are Verisign, Comodo, Godaddy, and Starfield. Please check the latest Intel AMT documentation for the most current list of partner CAs. The IT administrator obtaining a certificate from one of these commercial partner CAs will have to pay some fee for the certificate.

One important point to note is that the self-signed certificate generated by Intel AMT is not important to establish the authenticity of Intel AMT to the configuration server. Some internal design considerations in Intel AMT caused us to designate the configuration server as the TLS client and Intel AMT as the configuration server. The TLS protocol mandates that the TLS server must have a certificate. Hence Intel AMT uses the self-signed certificate only to provide confidentiality of the TLS channel. The authentication of the configuration server is established by the involvement of the certificate belonging to the configuration server. Remember, the main intent of securing the configuration channel is to ensure that a legitimate configuration server configures the Intel AMT subsystem. Therefore, the TLS protocol's main intent is to verify the identity of the configuration server. An optional mechanism (based on a One Time Password) exists to verify the identity and authenticity of the Intel AMT subsystem, which we will cover later on.

Another important aspect of Intel AMT configuration using asymmetric key based TLS protocol is that the Intel AMT firmware image comes pre-configured with cryptographic hashes (SHA1 hashes) of the root certificates of partner CAs. We will refer to these hashes as *certificate hashes* (or cert hash in short) in the rest of this chapter. Since these cert hashes are part of the firmware image, they provide a pre-configured and strong root of trust (that is anchored in hardware) for verifying configuration server certificates. The only reason why we chose to store a hash of the root certificate and not the entire root certificate itself is to save storage space on the nonvolatile flash memory. A full certificate could run into a couple of kilobytes of space, whereas a SHA1 hash of the certificate requires just 20 bytes for storage.

With this much preparation at both ends, we can now dive into the actual mechanics of the configuration protocol. To begin with, the configuration server and Intel AMT initiate the establishment of a TLS session between them. The following checks are done by Intel AMT to ascertain the authenticity of the configuration server as described in the sections below.

- Certificate chain validation
- Configuration server identity validation
- Certificate usage validation

Certificate Chain Validation

During the TLS session establishment, since Intel AMT offers a self-signed certificate, the configuration server cannot verify it. So the configuration server skips the verification of the self-signed certificate. Remember, we will not use the self-signed certificate for establishing trust anyway. The configuration server must send the entire certificate chain (including the root certificate) to Intel AMT during the TLS handshake. Since the configuration server certificate is signed by a CA whose root cert hash is already available to Intel AMT, Intel AMT can verify the entire certificate chain, first by ensuring that the root certificate is authentic (by calculating its hash and comparing it with the one stored in the firmware image). Subsequently Intel AMT verifies the rest of the certificates in the certificate chain following the validation procedures described in Section 6.1 of RFC 5280.

Now Intel AMT knows that the certificate chain is valid. It still does not know, however, whether the FQDN of the configuration server is something it should trust or not. For instance, if the Intel AMT machine actually belongs in a enterprise called foo.com, how does Intel AMT know that it should not get configured by a configuration server that presents a certificate with an FQDN of say, hacker.com? Remember, the legitimate owner of the domain hacker.com can get a legitimate certificate from a commercial CA for hacker.com. But we don't want this certificate to be used to configure machines in foo.com. Let us fix this problem by doing the configuration server's identity validation.

Configuration Server Identity Validation

To verify the authenticity of the FQDN presented in the configuration server certificate, Intel AMT obtains the DNS domain (the DNS suffix of FQDNs of machines in that DNS domain) of the network it is in by querying the DHCP server with option 15. Per the DHCP options RFC (RFC #2132), this option specifies the domain name that client should use when resolving hostnames via the Domain Name System. If in a particular network configuration, the

DHCP infrastructure does not support option 15, then this mechanism of configuration will not be available.

Now the attacker cannot use his certificate belonging to hacker.com to configure Intel AMT machines in the foo.com network because the DHCP and DNS information revealed by the network to Intel AMT will indicate the network DNS suffix as foo.com and not hacker.com. So we have made the job of the attacker more difficult. Now the attacker must also attack and defeat the DNS and DHCP infrastructure of the enterprise before it can attack Intel AMT configuration. This is a very significant barrier for the attacker to overcome.

Certificate Usage Validation

A potential problem still exists that needs to be fixed. An attacker can try to find a weakly protected certificate and private key somewhere else on the enterprise network that has a FQDN value that belongs to the enterprise. Then the attacker does not have to defeat the DNS and DHCP infrastructure. He would simply use this stolen certificate and private key and use it as an Intel AMT configuration certificate and private key. As an example, a weakly protected certificate and private key could be available in some old web server that someone hosted in the intranet, and later discarded.

Intel AMT first confirms that the certificate includes both "TLS Client" and "TLS Server" roles. This guarantees that the CA issuing the certificate has verified that the certificate applicant is associated with the name appearing in the CN field, as is the case for standard TLS certificates used on the Internet. Then Intel AMT also checks a particular qualifier in the certificate. This qualifier specifically denotes that this certificate is meant for Intel AMT configuration. Intel AMT will not accept any certificate that does not have this qualifier. So now the stolen certificate just became useless. We do not expect any old web server certificate to have this newly defined qualifier, thereby rendering any such old certificates unusable for purposes of Intel AMT configuration. When an IT administrator requests a certificate from a partner CA for Intel AMT configuration, he must explicitly require this qualifier, otherwise the certificate won't be usable for Intel AMT configuration. The CA would not add this qualifier for any other certificates (such as simple web server certificates). The qualifier can be of one of two types in any given Intel AMT configuration certificate.

- A well defined usage OID. This OID is defined and registered as an OID to denote Intel AMT configuration certificate. The OID value is 2.16.840.1.113741.1.2.3.
- A well known string in the OU field. The string is "Intel(R) AMT Setup Certificate".

Figure 17.5 shows two sample Intel AMT certificates with these qualifiers. The one on the left is issued by Comodo with the Enhanced Key Usage OID (highlighted). The one on the right is issued by Verisign with the well-known string in the OU field (highlighted).

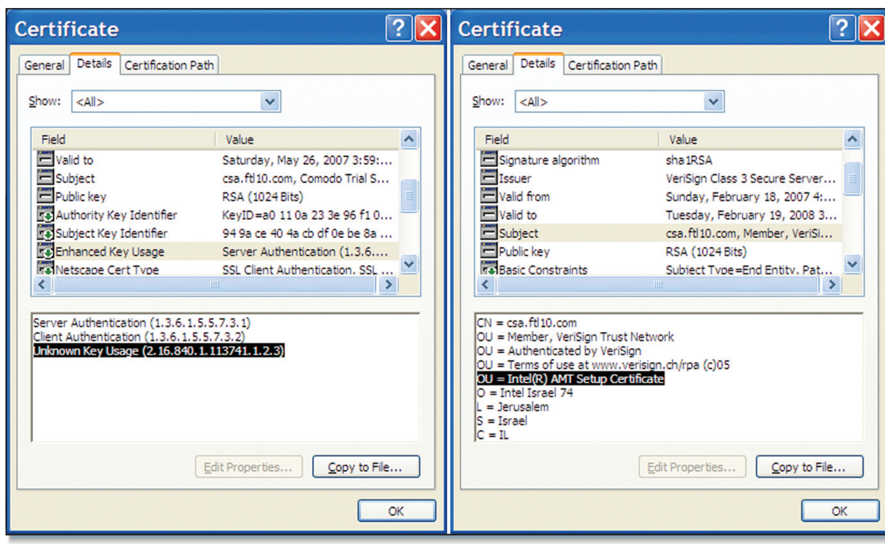

Figure 17.5 Intel® AMT Configuration Server Certificates with the Special Qualifiers

More Security Protections for Remote Configuration

Let us make the job of the attacker even harder. Let us assume that the attacker has successfully defeated the DNS and DHCP infrastructure of the enterprise. By the way, if this happens, then the enterprise is in deep trouble anyway, with or without Intel AMT in the picture. But since this book is about Intel AMT, we will not speculate how else could the attacker harm the enterprise. We will restrict our focus to Intel AMT only.

Configuration Enabling from a host OS based ISV agent When Intel AMT is in the unconfigured state, it keeps its out-of-band (OOB) network interface closed1. So an attacker cannot even communicate with Intel AMT, let alone attempt to configure it with illegitimate means. So even a compromised DNS and DHCP infrastructure does no harm to Intel AMT at this point. The only way for Intel AMT to open up its OOB network interface is for a software agent on the local host OS of the Intel AMT machine to tell Intel AMT to open its OOB interface. A software agent sends this command to Intel AMT via the local host to the Intel Management Engine (Intel ME) communication channel called Intel Management Engine Interface (Intel MEI), which we have covered in Chapter 7. So, to add to the attacker's woes, in addition to defeating all of the above barriers, he now also has to somehow sneak in a malicious software agent on all the Intel AMT machines. This malicious agent must escape undetected by the anti-virus and anti-spyware software on the host OS of the machine. This is therefore not an easy attack point either.

In the normal scenario, the IT management consoles and agents need to be augmented to push a script or piece of software (such as, for instance, an ISV agent) on the local host OS from a centralized IT management console that issues the command to Intel AMT to open up its OOB network interface and starting the configuration process.

One Time Password Another thing that the ISV agent (as discussed in the previous section) does is that it sets a one time password (OTP) into Intel AMT via the MEI. This is called an OTP because this password is never used later on after the configuration process is over. The OTP is recommended to be unique for every Intel AMT computer. During the actual configuration process over the OOB network interface, the Intel AMT subsystem needs to return this OTP to the configuration server as a proof that this is the same Intel AMT subsystem that the configuration server wants to be configured. The OTP flow from the ISV management server to the ISV agent, to the Intel AMT subsystem, to the configuration server, and back to the ISV management server helps to establish the chain of trust at the management server side, especially that the entity being configured is the same one that the management server wanted to configure.

1 Some Intel AMT will keep the out-of-band interface open for a few hours after first power own. Check OEM manuals for details.

If you notice carefully, so far there has been no need to manually touch the Intel AMT machine at all, either by the OEM or by the IT technician. All the operations above were based on communication between Intel AMT and a remotely placed configuration server on the network. All the technician had to do was to purchase an Intel AMT configuration certificate from a partner CA and install it into the configuration server. Also, the technician had to send a software update to the Intel AMT machines (via already defined software update mechanisms) that would send a command to Intel AMT to open its OOB interface. We call this mechanism of Intel AMT configuration Remote Configuration.

One Touch Enhancements to Remote Configuration

The barriers offered by remote configuration are significantly high for the attacker to break or circumvent. But, believe me, some enterprises are extremely cautious regarding the security of their IT infrastructure. Or, some enterprises have IT policies that they need to comply with, and the Remote Configuration protocol needs to be further enhanced to comply with those policies. For such enterprises, the security of the configuration protocol can be further strengthened so long as they are willing to touch the Intel AMT machines and configure some parameters into the Intel AMT subsystem. The mechanisms to configure these parameters via a one-touch operation still remain the same as described earlier, namely:

- Manually via the Intel MEBX interface
- In a automated manner by using a USB thumb drive via the Intel MEBX interface
- By negotiating with the OEM to configure the parameters at the time of manufacturing the machine

The following two aspects are optional and further increase security by requiring a touch to the Intel AMT computer.

Enterprise Root CA Suppose an enterprise policy disallows trusting a commercial CA for root of trust. The policy only allows for enterprise's own root of trust. To solve this problem, Intel AMT allows the use of an enterprise CA root of trust instead of trusting a partner commercial CA such as Verisign. The enterprise IT manager can decide to place the cert hash of the enterprise root certificate in the nonvolatile flash memory of the Intel AMT subsystem (using one of the aforementioned one-touch operations). If this cert hash is placed in the nonvolatile flash and the pre-configured commercial CA cert hashes are marked disabled, then Intel AMT will use only this cert hash for purposes of verifying configuration server certificates. The configuration server will then have to use a certificate issued by this enterprise CA, instead of any other CA. This mechanism completely cuts off an attacker from being able to attack the configuration process of Intel AMT because the attacker needs to have access to the CA infrastructure of the enterprise, which is usually a very heavily guarded asset.

Trusted Pre-Configured DNS Suffix Now suppose that an enterprise finds it acceptable to trust a partner commercial CA, but it does not trust its own DNS and DHCP infrastructure to be very secure. This could happen for many reasons, such as outsourced management (or management by contractors) of DNS infrastructure, or shared DNS infrastructure across sister organizations of a larger parent enterprise. Or possibly the DNS domains are so disorganized that relying on DNS domain suffixes obtained from DHCP option 15 is not practical. Or maybe that DHCP option 15 is not supported in a given environment. How does Intel AMT verify the domain suffix to compare against the FQDN in the certificate? To overcome this problem, Intel AMT allows a domain suffix to be configured into the nonvolatile flash memory of the Intel AMT subsystem (using one of the aforementioned one-touch operations). If this domain suffix is configured into the nonvolatile flash, then Intel AMT does not use DHCP option 15 to learn the network's domain suffix. Instead it uses the domain suffix configured using the one-touch operation for validating the FQDN in the configuration server certificate.

Remote Configuration Protocol Summarized

Armed with the understanding from previous sections, let us look at the protocols used for remote configuration of Intel AMT. We break down the protocol into two phases as shown in Figures 17.6 and 17.7.

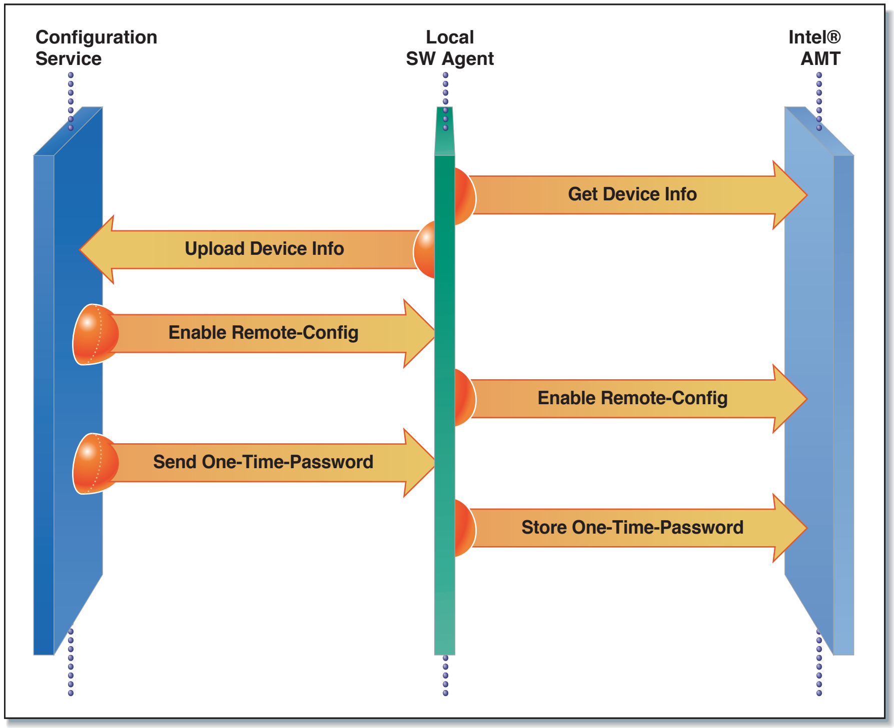

Figure 17.6 Phase 1 of Intel® AMT Remote Configuration

Remote Configuration: Phase 1

To start with, Intel AMT is unconfigured, its network interface is disabled, effectively disabling any remote configuration attempts; and its MEI is enabled. In Phase 1, a local host ISV software agent detects the state of Intel AMT by using the MEI, and then can upload device information to the configuration server. The information includes the firmware version, the installed root certificate hashes, and whether the device is configured to operate in PSK or Asymmetric Key provisioning mode. In this scenario we assume it is the latter.

Next, the configuration server instructs the agent to enable remote configuration of Intel AMT, and it provides the agent with the OTP. Once Intel AMT enables remote configuration and stores the OTP into Intel

AMT, it enters Phase 2 of the protocol. In phase 2 the configuration server communicates directly with Intel AMT via the OOB channel, as shown in Figure 17.7.

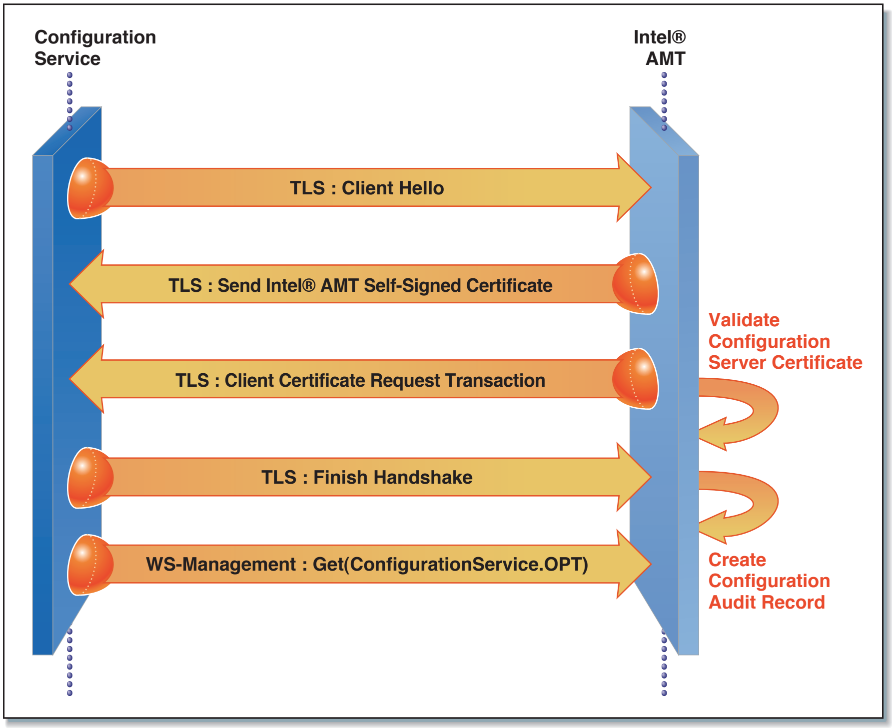

Figure 17.7 Phase 2 of Intel® AMT Remote Configuration

Remote Configuration: Phase 2

In Phase 2 of the configuration flow, the configuration server and Intel AMT establish a TLS session between them. This phase commences when the configuration server opens a TLS session with the Intel AMT device. As part of the TLS handshake, Intel AMT sends its self-signed certificate. The configuration server skips the verification of the self-signed certificate. However, the server requires the public key embedded in it to successfully complete the TLS handshake with Intel AMT. Next, the configuration server sends the entire TLS certificate chain (including the root certificate) to Intel AMT. Intel AMT performs validation of the configuration server by validating the information provided in the certificate chain. The validation includes validating the following aspects as discussed earlier.

- Mandatory: Certificate chain validation
- Mandatory: Configuration server identity validation
- Mandatory: Certificate usage validation
- Optional: Validation of the certificate chain rooted in the enterprise root CA
- Optional: Validation of the server identity against a trusted pre-configured DNS suffix

The TLS session is now established.

Once the TLS connection is established, the configuration server requests the OTP from Intel AMT. The configuration server validates the OTP against the one it had stored for this Intel AMT device. If the two match, the session setup is completed, and the configuration server proceeds with the configuration of the Intel AMT device with the enterprise specific information.

Configuring Enterprise Data

The configuration of the Intel AMT device entails downloading a bundle of enterprise specific information (such as access control policies, wireless profiles, and security parameters) that allows the Intel AMT functionality to operate properly within the enterprise environment. Some of the pieces of information that are downloaded into Intel AMT as part of the configuration process are as follows. This is not a complete list however.

- Network policies (TCP/IP settings, DNS settings, and so on)
- Active Directory policies (Kerberos settings, master key, and so on)
- Current Time (for PRTC)
- Wireless profiles
- 802.1x profiles
- TLS settings (enabled/disabled, private keys and certificates, cipher selection, and so on)

- Administrator usernames, passwords (HTTP digest), access control list
- Audit Log policies, auditor authentication credentials

Intel AMT leverages the CIM of the DMTF to represent the various configuration settings that are communicated to Intel AMT. This configuration is done over the WS-Management protocol. Certain configuration properties of Intel AMT utilize DMTF's management profiles mandated by DASH. For example, local user-account management and authorization are based on DMTF's Simple Identity Management and Role Based Authorization profiles. The Intel AMT SDK provides a complete list of supported management profiles.

Configuration Audit Record

Once the Intel AMT device establishes trust with the configuration server, it creates a configuration audit record, recording the configuration TLS certificate details and additional parameters. This record is subsequently locked down to prevent any further modifications, but it is still available for being read via the local MEI interface as well as through the Intel AMT network interface. Since the record is "read-only" it allows policy enforcement applications to detect occurrences of un-authorized configuration and use of Intel AMT systems.

Bare-Metal Configuration

The scenario described earlier in the chapter is called the "Delayed Configuration" scenario, and the assumption is that the Intel AMT configuration process takes place once the host OS is already deployed. Recall that Phase 1 for the previously mentioned configuration method required a software agent to enable the network interface of the Intel AMT system and provide discovery information back to the configuration server.

Bare-metal configuration is another configuration capability of Intel AMT that allows configuration prior to OS installation. In fact one key usage of bare-metal configuration is to push down an OS installation or image to the platform by using Intel AMT remote boot operations. Naturally, this step can only take place after Intel AMT has been configured.

Both Pre-Shared Key and Asymmetric Key methods can be utilized for bare-metal configuration. Intel provided the system manufacturers with the capability to designate in manufacturing a "bare-metal timer," typically limited to 24 hours, in which Intel AMT enables its network interface. This allows a configuration server to configure a device without the need for the software agent trigger, required in Phase I of the "Delayed Configuration" model. Bare-metal configuration is enabled from the initial boot of the system and for the accumulated system up-time duration, specified by the bare-metal timer. After this duration, Intel AMT disables its network interface. To configure Intel AMT from this point onward, a delayed configuration method must be used. Figure 17.8 is a sequence diagram depicting Phase 1 for bare-metal configuration. In order to use bare-metal configuration, an alias for the configuration server address is registered on the relevant DNS servers in the enterprise. This allows Intel AMT to discover the IP address of the configuration server by querying the DNS server.

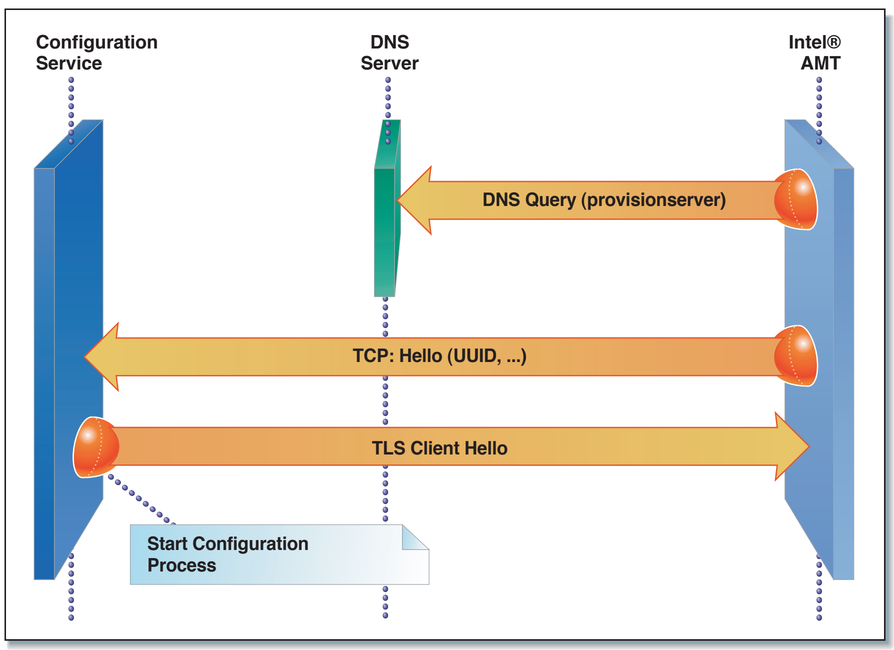

Figure 17.8 Bare-Metal Configuration of Intel® AMT

During the bare-metal time window, Intel AMT tries to acquire a DHCP IP address, detect the DNS server address, and use that address to query for the designated configuration server. Intel AMT uses a concatenation of a predefined host name “provisionserver” and the DNS suffix it has learned, to query DNS for an IP address. Intel AMT sends a notification to the configuration server, depicted in Figure 17.8 as a “Hello” message. The Hello message is TCP-based and provides the configuration server with the platform’s additional information that can assist in completing the configuration process. As long as the bare-metal time window has not elapsed, the configuration server can initiate a TLS session with Intel AMT (by using either the PSK or the Asymmetric Key method), and complete the configuration process.

Summary

In this chapter we discussed the various scenarios that are supported for configuring Intel AMT, the various mechanisms and protocols available for configuration of Intel AMT, and we outlined the various options and parameters that can be adjusted to make the tradeoffs between security, cost, and convenience. One of the aspects of Intel AMT configuration that we want to reiterate here is that Intel AMT offers a wide selection of configuration options, catering to almost every type of customer, ranging from a small home business to a Fortune 500 enterprise. At one end of the spectrum, it is possible to configure Intel AMT in a matter of minutes, and get it up and running on a test machine. At the other end of the spectrum, it is possible to configure a vast array of Intel AMT machines in a large enterprise, without even physically touching those machines once (remote configuration); moreover, they can be configured in such a way that the process is trusted and secure, and not vulnerable to being attacked or snooped by malware or other prying eyes. Several optional parameters can further increase the security strength of the remote configuration process. It is also possible to remotely configure Intel AMT on a bare metal computer (that is, a computer with no operating system installed).