Spectralink 84-Series Security Best Practices

The Spectralink 84-Series handsets leverage a proprietary operating system, but this also carries with it the need to understand how to properly secure and manage your systems. In this document, we'll cover the commonly known and the not so commonly known security topics you need to know.

Security Primer

Let's start with some basics of security and the terminology that will be used. We want to make sure this content is understandable and that requires a foundation. Without going so deep as to teach everyone who reads this the OSI Model[1] we still want to provide enough detail to make the content clear. Since everything we will discuss in this document starts with protocols, we will start by providing a high level overview of the supported protocols and what they each mean. Please keep in mind that this document is not meant to be your exclusive source for knowledge, but instead just a taste to get you started. We encourage you to seek out more information and continue your education on this topic.

Protocols

The 84-Series handset is your typical wireless device in most situations and, as such, it supports the typical transport protocols such as UDP, TCP and TLS. But there's more to protocols than just transport. The following table is a list of just some of the protocols the 84-Series handset supports and a brief explanation of what they do. This is not meant to be an exhaustive list. There are also different versions of protocols, which we will get into later. We'll put a more complete list of protocols in Appendix A at the end of this document.

Protocol	*Description*
TCP	Transmisson Control Protocol – One of the most basic of transport protocols. This connection-oriented protocol handles the delivery of user data packets via a more managed delivery method via checks and balances to make sure the traffic arrives at the destination.
UDP	User Datagram Protocol – The most basic of transport protocols. This connectionless protocol is essentially a "send and forget it" protocol that doesn't rely on any error correction or checks to ensure delivery like TCP.

[1] https://en.wikipedia.org/wiki/OSI_model

Protocol	*Description*
TLS	Transport Layer Security – TLS is the more complex brother of TCP. TLS uses much more secure methods to not only perform traffic delivery, but also to provide methods for traffic encryption.
SSL	Secure Socket Layer – The predecessor of TLS and a protocol that was primarily used in HTTPS transactions. SSL is depecated in nearly all environments now due to security flaws, more on that later.
HTTP & HTTPS	Hypertext Transport Protocol & Hypertext Transport Protocol Secure – any time you open a web page via your internet browser, you're going to be using either of these two protocols. Most likely you will use HTTPS as it will then provide a secure connection via TLS to the site you're connecting to via a certificate issued to that site.
FTP	File Transfer Protocol – FTP has been around for a long time and is used in the most basic of file transfer situations. FTP allows for the transfer of large numbers and sizes of files while still providing a means to authenticate the user before permitting access to the content. The connection is not secure, nor encrypted though which allows for someone in the path of that transfer to see both the username and password transmitted in clear text along with every file sent.
SFTP	SSH File Transfer Protocol – SFTP is essentially the same thing as FTPS except that it takes advantage of a system that has SSH (Secure Shell) enabled for remote access. Secure Shell allows a user to communicate with the destination endpoint over an encrypted connection using a key which will ensure the user credentials and all files are sent encrypted.
FTPS	File Transfer Protocol Secure – FTPS is not all that different from SFTP except that it uses TLS, originally it used SSL, to communicate with the destination endpoint. Otherwise, the connection is encrypted in much the same way it is with SFTP ensuring that user credentials and files are all encrypted in transit.
TFTP	Trivial File Transport Protocol – This is the most basic of file transfer protocols. It is not encrypted and does not require any sort of user authentication to access the files stored on the destination TFTP server. It is not recommended for use by anyone that cares about security even a tiny bit!
SIP	Session Initiation Protocol – SIP is actually an offshoot of HTTP in how the messages look and how they're built. By default it is insecure, but also doesn't contain a great deal of information that could be used unless someone knows what they're after and what they're doing, e.g. someone looking to attack your SIP server.
SIPS	Session Initiation Protocol Secure – SIPS is the same as SIP, but with the added layer of security added to it. This extra security is provided by TLS encryption using certificates and changes how the SIP URI header looks. It also requires that the communication remain peer-to-peer or the secure connection will break down.
RTP	Real-Time Transport Protocol – When making voice calls, RTP is your audio traffic. This traffic is encoded using an audio codec, but the traffic itself is not encrypted and can therefore be listened to by someone able to intercept this traffic.

Protocol	*Description*
SRTP	Secure Real-Time Transport Protocol – When you need to ensure your voice traffic is encrypted, you must use SRTP. This protocol uses TLS to encrypt the traffic via certificates and is typically done bi-directionally to keep the entire conversation secure. Nearly all SIP devices will support SRTP, but it must be negotiated during call setup and will require matching ciphers on both ends. We'll get into ciphers later.

The Spectralink 84-Series handset runs on a proprietary operating system which is based on a Linux kernel. As many people likely know, the handset was originally developed while Spectralink was affiliated with Polycom© and is based heavily on the VVX1500® software. If you're familiar with that desk set then you'll be familiar with much of the 84-Series handset. The major differentiator should be fairly obvious though, the wireless connectivity. This introduces a lot of additional complexity that you won't find with a wired ethernet connection. Since this is a significant portion of what makes Spectralink products special, let's spend some time on that area next.

Before moving on, let's cover one more important topic. What defines a secure connection?

Many people think that a secure connection only requires that a username and password be used. But the reality is that unless some sort of encryption is used, the connection is quite insecure. Encryption requires that a mutually agreed upon key be used between both ends of the connection being created to allow for the data sent to be encrypted and decrypted with this key. Keys often take the form of certificates because there is a public and private side to the key and they can be used for many different connection types. Other keys can be calculated using a pre-shared value and a random value so that both ends can determine the final key based on the same information. So just remember that a connection isn't truly secure without some form of encryption present.

Wireless Security

Wireless security, for many people, is a bit of a black box. There are many options and some are much more complex than others. This often leads administrators to choose security based on how easy it is to maintain and deploy. Not always the best choice, but obviously the easier one given all the other considerations for a voice platform. To try and help simplify and demystify the wireless black box of security let's spend some time discussing the different security mechanisms and how each one has evolved and can be used with the Spectralink 84-Series handsets. We'll also talk about which ones best suit certain deployments and why to try and help you make the right decision for your own deployment. Keep in mind that this isn't meant to make the decision for you, but to help you determine what's right based on your unique business needs, requirements and what your solution needs to provide to your users for the best experience. One of the best ways to approach this will be to tackle the different security options from least secure to most.

WEP (Wired Equivalent Privacy)

When wireless communications, specifically 802.11 without all the alphabet soup, were initially available there was a need for encryption that provided some level of protection. With WEP, which was ratified in 1997, the administrator could deploy a solution that provided sufficient protection of the user's data that people felt comfortable using wireless networks. However, as with so many things, this didn't last very long. By 2004, WEP was deprecated in favor of better solutions that provided better security. At this point in time, no one should be using WEP. Many platforms are even removing support for WEP as it is easily accessed through a variety of methods. At this point, Spectralink still supports WEP, but we do not recommend that anyone use it for any reason.

WPA (Wi-Fi Protected Access)

As wireless technology progressed, and hackers found ways to access prior security solutions, the need for better solutions became obvious. In 2003, the Wi-Fi Alliance introduced WPA as a better option for security wireless communications. This solution uses TKIP (Temporal Key Integrity Protocol) which apply per packet keys instead of the manually entered key that WEP used. This solution provided a way to perform message integrity checks (MIC) which was better and less computationally impactful than those in WEP. But there were also some fatal flaws in WPA that made it susceptible to re-injection and spoofing. WPA, while still available and supported by most platforms, is also not recommended for use. In order to meet requirements for HIPPA, PCI and other government standards, TKIP just won't cut it.

WPA2 (Wi-Fi Protected Access 2)

The Wi-Fi Alliance introduced WPA2 was ratified in 2004 to correct the vulnerabilities identified in WPA with TKIP. The biggest difference between WPA and WPA2 is the mechanism by which it secures the client data. With WPA2, this is required to be done via CCMP with AES. CCMP is a beast all its own, Counter Mode Cipher Block Chaining Message Authentication Code Protocol, which uses 128-bit keys and a 128-bit block size and is all based on AES processing. We won't get into the details of CCMP or AES here as it's far to complicated to bother with, but suffice to say is that it has been the go to security solution for the last 16 years. WPA2 support has been a requirement for Wi-Fi Alliance certification since 2006 and while it can be decrypted if you have the network SSID and a copy of the pre-shared key or passphrase used, and capture the handshake during boot up, it is still considered one of the best solutions for wireless security. That is, primarily for home users rather than enterprise environments. Enterprise environments have, and still do, use WPA2 extensively given its ubiquitous support and level of security provided. For most environments, this is going to be fine, such as non-healthcare, financial, retail or other enterprise environments where government regulations do not require greater security. When meeting government regulations, you're going to want an enterprise level security option. We'll cover those in a moment.

WPA3 (Wi-Fi Protected Access 3)

In 2018, the Wi-Fi Alliance announced WPA3 as a replacement for WPA2. As the newest flavor of WPA, this standard uses stronger encryption when in enterprise modes. We'll cover enterprise modes next. The standard approach of a pre-shared key gets replaced and instead uses a method called SAE (Simultaneous Authentication of Equals). This process is quite different than prior exchanges. It still uses a password that both the client and network/AP have in common, but that password is never sent. Instead, both the client and network use the password and the MAC addresses of both peers to generate a cryptographically strong key that will be used. This approach still uses CCMP with AES when in personal mode as the minimum encryption method, but strong options are still available. Spectralink will support WPA3 only on Android platform device running Android 10 or later versions in a future release of software. Given the limited infrastructure support for WPA3 and the slow adoption rate, there is currently no market driver to increase the time to market for this support. The Android platform already has support for WPA3, but to ensure seamless performance for voice, Spectralink has opted to limit access to WPA3 for now.

Enterprise Security

Within Enterprise security, there are a number of different components that we'll cover. Since Enterprise security is offered in WPA, WPA2 and WPA3, it only makes sense to combine all of them into a single section. The primary driver for Enterprise security is to provide greater security while adding support for authorization and accounting. The additional components make it possible for administrators to manage their environment more granularly and provide documentable information should a breach or other security incident occur that requires reporting. Enterprise security is also a must have for nearly all government regulatory bodies when it comes to HIPPA, PCI and many others. When you read into the details of these regulations, most will not specify a specific security methodology and will instead state a need to "*…reasonably and appropriately protect the confidentiality, integrity, and availability of*[2]*…*" any information that is created, received, maintained or transmitted. In order to achieve these regulatory requirements, you're going to want to implement a security standard for your organization that exceeds the expectations of the law. Keep in mind that because these regulations are intentionally vague and they leave a lot of room for interpretation. But when a breach of security happens and information is leaked, you have to remember that these regulations become much less vague and instead of sharp points that can lead to significant penalties. HIPPA alone has the potential for individual penalties, including jail time for those individuals involved in a breach of information if there were things those people could have done to prevent the incident.

Within Enterprise security there are a number of different options. They all revolve around a central protocol called EAP, Extensible Authentication Protocol. EAP is an authentication

[2] https://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/administrative/securityrule/pprequirements.pdf?language=es

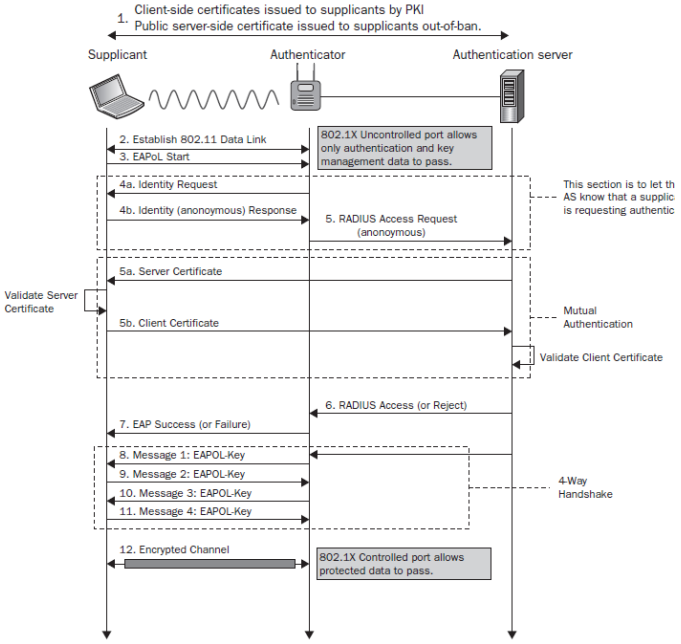
framework that is intended to be used by network and other internet connections. For authentication to be possible you will also need to have RADIUS (Remote Access Dial-In User Service) system that will accept the authentication request. We'll talk about RADIUS more later. Enterprise security offers a lot of different authentication solutions. The following table describes a few of the primary EAP types and denotes on which Spectralink platforms they are support.

EAP Type	*Description*	*Versity (95/96 Series, 92 Series) Support*	*84-Series Support*
PEAP	Protected Extensible Authentication Protocol – protocol that encapsulates EAP within an encrypted and authenticated tunnel. Typically includes MSCHAP (Microsoft Challenge Handshake Authentication Protocol) to provide password management and authentication to Microsoft Active Directory.	X	X
TLS	Transport Layer Security – protocol that uses X.509 digital certificates for authentication via mutual authentication mechanisms whereby client and server are both authenticating each other.	X	X
TTLS	Tunneled Transport Layer Security – extension of EAP-TLS that allows for the client to optionally be authenticated. Note that this is a less secure option that TLS.	X	
PWD	Password – An EAP method that uses a shared password for authentication.	X	
SIM	Subscriber Identity Module – Used for authentication and session key distribution using the SIM from a cellular network.	X†	
AKA	Authentication and Key Agreement - Mechanism for authentication and session key distribution using the UMTS USIM	X†	
AKA'	Authentication and Key Agreement Prime - Variant of EAP-AKA used for non-3GPP access to a 3GPP core network.	X†	
FAST	Flexible Authentication via Secure Tunneling – A Cisco proprietary authentication method leveraging PAC (Protected Access Credential) files, similar to certificates, for authentication.		X

† Only supported on the Versity 96-Series when a SIM card is inserted that supports the specified security method.

Enterprise security relies on one of the above EAP types, associated with WPA, WPA2 or WPA3 in Enterprise mode, e.g. WPA2-Enterprise with EAP-TLS. This would mean that the client device is configured with the security type WPA2-Enterprise and the authentication type selected would be EAP-TLS. For the purposes of regulatory compliance, EAP-TLS is the

preferred authentication method. It more secure than the other methods above as it provides a means for the client to authenticate the server's certificate and for the server to authenticate the client's certificate thus creating mutual authentication and more secure tunnel. Additionally, instead of sending the traditional username and password like you would with PEAP authentication, the certificate issued to the client will act as the credentials. This eliminates the transmission of any data being sent in the clear as secure tunnels are created during the handshake process after the certificate exchange, also called Server and Client Hello. Figure 1 is a visual description of the EAP-TLS authentication process. If you were perform a wireless packet capture of a client attempting EAP-TLS authentication, it would look something like this.



1. Client-side certificates issued to clients by PKI, Public server-side certificate issued to clients out-of-band

a. The client and the authentication server begin by saying "hello" and prepare their certificates for authentication to establish a trusted connection.

2. Establish 802.11 Data Link

a. The client establishes a connection to the authenticator. This will allow for a secure exchange of information between the two parties.

3. EAPoL Start

   a. EAPoL (Extensible Authentication Protocol over LAN) indicates that information can be exchanged between all three parties over a secured LAN channel. Additionally, this is where the authentication method is determined – in this case, EAP-TLS.

4. Identity Section

   a. Identity Request

      i. The client requests the identity of the authenticator to ensure it is sending the client certificate to the correct place.

   b. Identity (anonymous) Response

      i. The authenticator requests that the client identify itself.

5. RADIUS Access Request (anonymous)

a. The information that identifies the client and authenticator is sent to the RADIUS to confirm their identity and allow for authenticating information to be sent.

b. Server Certificate

    i. The RADIUS sends its server certificate to confirm its identity through server certificate validation

c. Client Certificate

    i. The client validates the identity of the authentication server certificate. After validation, the client sends its client certificate.

6. RADIUS Access (or Reject)

    a. The RADIUS authentication server receives the client certificate and authenticates its identity as an approved network user. Depending on the user's certificate, the RADIUS sends an Access or Reject message to the authenticator.

7. EAP Success (or Failure)

    a. Based on the RADIUS Access or Reject message, the authenticator sends a Success or Failure message to the client to indicate whether they have been approved or denied network access. If the message is Success, the network path is opened for direct network communication between the client and authentication server.

8. Message 1: EAPOL-Key

9. Message 2: EAPOL-Key

10. Message 3: EAPOL-Key

11. Message 4: EAPOL-Key

    a. The next step is a series of messages known as the EAPOL-Key exchange. It is a 4-step handshake between the authenticator and client that generates encryption keys. These keys are used to encrypt information that will be sent over the wireless connection and ensures that all ongoing network communications are encrypted and cannot be read by outside parties.

12. Encrypted Channel

    a. The end result of EAP-TLS authentication is an encrypted channel of communication. The user is ready to access the secure network and utilize all resources available to them.

Provisioning

Now we will shift gears a little bit and discuss the provisioning methods used by the 84-Series handset. This will highlight the secure and insecure provisioning methods and why you might choose one over another. We will also discuss the data at rest, which is the handset configuration files and how they are stored in both the provisioning server and the handset.

Insecure Provisioning Methods

There are a few options when it comes to insecure provisioning methods. From least secure to most are, TFTP, HTTP, and FTP. TFTP is highly insecure as it does not require any type of authentication. There's also no way to encrypt the data to make it safe for transmission. Anyone that were able to capture packets wired or wirelessly would be able to see everything being transmitted. HTTP isn't a lot better than TFTP, but it does at least permit the use of credentials. You could also use a secure version of HTTP, but we'll talk about that in a minute. The other limitation of HTTP is that you cannot append data to logs. There is only the ability to replace logs on the provisioning server. This leads to lost and missing data. Lastly, is FTP. FTP requires the use of credentials but does transmit the data in the clear. This means that anything you transmit can be viewed with a packet capture. But it does at least offer the ability to provide greater restrictions to the contents on the provisioning server without completely compromising the security of your data like you would with TFTP.

Secure Provisioning Methods

Secure provisioning of the handset is really only possible when using both credentials and encryption of the data in transit. The two protocols available to accomplish this are HTTPS and FTPS. These are listed in order of preference of use. HTTPS does have the ability to use credentials and will require the use of a certificate being installed on the provisioning web server. The phone would then need to have either the root certificate of the CA that signed the web server's certificate or if it is a self-signed certificate, then just the web server's certificate. Just as with the HTTP server, there is a major limitation with HTTPS where the handset is unable to perform append for log files. It can only replace log files on the server, which can lead to lost and missing data.

FTPS is essentially the same as FTP, but done on port 990 and requires a certificate be installed on the FTP server. Don't confuse this with SFTP, which is FTP over port 22. They are different. Just like with HTTPS, the certificate installed on the server is used to encrypt the data in transit to handset. The phone would then need to have either the root certificate of the CA that signed the web server's certificate or if it is a self-signed certificate, then just the web server's certificate. The most significant benefit of FTPS is that the handset has the ability to perform appends to log files and with a secure connection you can be more assured your data is safe from prying eyes.

Data at Rest

Something that most administrators don't think about is the idea of data at rest. What does your data look like when it is not actively be transmitted or processed by devices and systems? This

is a pretty significant concept and can actually make or break some compliance requirements. Fortunately, there are options available to you with 84-Series handset. When deploying on a traditional provisioning server, there is a tool available to encrypt the configuration files that the handset uses. The files are then encrypted at rest on the provisioning server and will be stored in the handset encrypted as well. You will define a key that is then shared with the handset during initial provisioning and the files stored on the provisioning server will be encrypted using this same key. You can obtain the application from the Spectralink Support site, but you will need a valid software maintenance agreement in order to access the download.[3]

Certificates

Certificates are an integral part of security used everywhere, and have been for years now. We could spend days just talking about certificates and all of the different aspects about them. We'll cover some of the key points around certificates and how they are used by the handset in different scenarios.

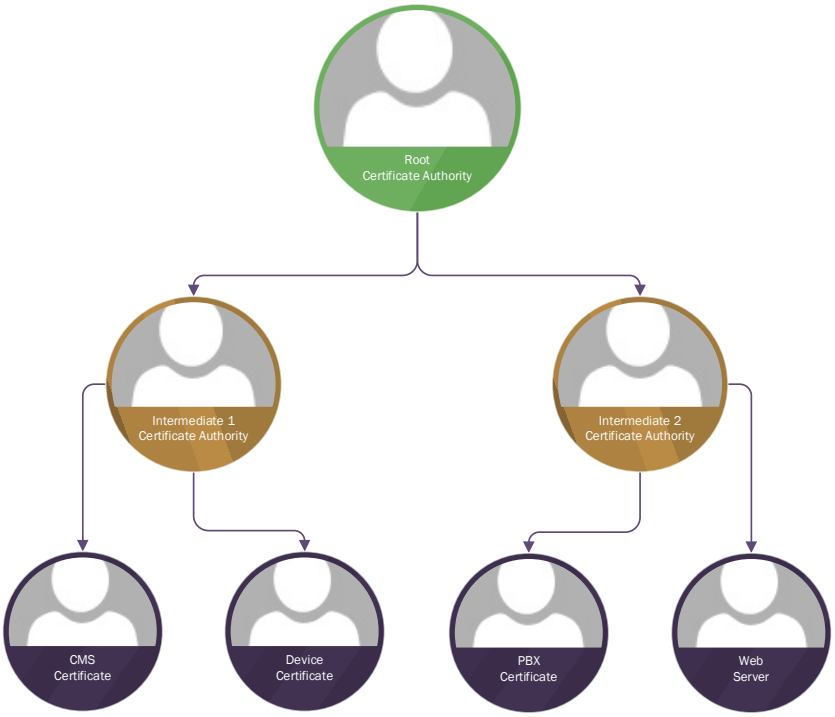
Certificates Overview

Probably the biggest aspect of certificates that is misunderstood is the chain of authority. Certificates are a bit like a chain of command. There's someone at the top (Root Certificate Authorities) that has all the authority and can delegate some of that authority to others (Intermediate Certificate Authorities). Then there are the ones at the bottom that are issued to them by those authorities above. Those certificates at the bottom have many different capabilities, or responsibilities that they can be empowered to perform.

Think of the Root Certificate Authority (CA) as the boss that is empowering his middle managers, the Intermediate CA's, to do the same tasks that the Root CA performs. This task is typically just signing certificate requests presented to them. If there is an Intermediate CA present, then the Root CA will likely never sign certificate requests to help preserve the chain of authority. There could be multiple Root CA's at the top, but they do not have responsibility over each other. They're essentially separate entities. But, Root CA's can, and frequently do, have multiple Intermediate CA's associated with them. That gives the system the ability to have any one of the Intermediate CA's sign certificate requests. But that also means that the chain of authority still follows whichever Intermediate CA to the Root CA ends up signing the certificate request.

All of that may seem a little convoluted and confusing. So let's look at it in a graphical format to try and make it easier to understand.

[3] https://support.spectralink.com/system/tdf/resource_files/ConfigFileEncryption_v1.0.zip?file=1&type=node&id=13105



From this diagram, you can see that the Root CA is at the top of the hierarchy. This certificate is a self-signed certificate, meaning that the Root CA is saying it has authority simply because it does. There are then two Intermediate CA's, which would have submitted Certificate Signing Requests (CSR) to the Root CA to be empowered with the authority to sign certificate requests like the Root CA does. This means that the Intermediate CA's are not self-signed and are now a part of the chain of authority. Finally, you have each of the certificates for the endpoints. Each of these would submit a CSR to the Intermediate CA for signing. These requests include information about the device making the request and the responsibilities that this endpoint wants to be empowered with. These responsibilities, also called Key Usage and Enhanced Key Usage, are what allow a certificate to sign other certificates, perform key encipherment, server and client authentication, and many more.

This all just scratches the surface of certificates. Things like the signature hash algorithm, cipher suites, public and private keys and so much more. There can also be more layers within the chain of authority where you might have Issuing Certificate Authorities between the clients and Intermediate CA's. There's really no limitation on the number of layers between the Root CA and the clients requesting certificates. And the names of the layers aren't really standardized anywhere but they do typically follow the same naming conventions.

The last few things we need to cover in this overview are around the endpoint certificates and their components for creation. These are the certificates that will play the biggest role in the further discussions in this document. Let's start with the CSR, since this is where all this information is initially provided. Every CSR has a number of required fields that help fill in a bunch of the values that get included in the certificate that is issued. All of this information is important, but some are moreso than others. First, is the common name (CN). This field is the hostname, FQDN (Fully Qualified Domain Name), or IP address of the endpoint requesting the certificate. These days, it is always recommended to use a hostname. This helps to ensure your certificates are assigned to specific endpoints that are also resolvable in a DNS (Domain Name Server) service. Tied to the CN is the Subject Alternative Name (SAN). The SAN is the same as the CN, except that it can have multiple entries and can include aliases for this certificate. That

means that if this endpoint has multiple names, or you want to be able to have this certificate be relevant for an endpoint that doesn't have the hostname in the CN field, this will allow you to still use this certificate. Something important to understand though, is that the SAN **must** include the value that is in the CN. If the SAN doesn't have the same value as the CN, there are multiple situations where this will cause certificate validation failures.

Last, we'll cover the concept of public and private keys. When a CSR is created, there is a private key that is created at the same time. This private key is defined based on some of the settings defined in the CSR. The key size is the biggest one and should always be at least 2048 at a minimum, but never more than 4096. The larger the key size, the longer it takes to process the certificate. Another is the certificate signature algorithm used. In years past, this was always SHA1 by default. But these days, the minimum should be SHA256 to ensure adequate key security. When the CSR is signed by the Certificate Authority, it is creating the Public Key, the actual certificate. This public key is what gets shared with everyone in the world. The Private key is kept by the endpoint and should never be shared. Combining the two is what permits things like key encipherment.

When addressing chain of authority though, the other endpoints that are trying to connect will need to have the Root CA and the endpoint, now a server because of the incoming connection, needs to send the entire chain of authority to the client for validation. And since the client has the Root CA, it will be able to look at the chain of certificates that the server sent and see that they are valid, thus allowing the connection to continue. This validation includes checking the Common Name and Subject Alternative Name fields in the certificate to make sure that they match the values for the server connection being requested. If they don't match, this can cause the connection to fail unless name validation is disabled.

Certificate chaining may seem like an odd thing to do since the certificates should be showing the signer. While it's true that the server certificate includes the CA information on which CA signed it, it doesn't always include all the certificates in the chain in this one certificate. Certificate chaining is done by opening each of the certificates in the chain of authority in a base64 format so that they are text readable. Then, starting with the server/endpoint certificate being first; you paste in each certificate in the chain up to the highest level of authority as the last certificate in the file. Each certificate must include the header and footer included in each certificate and there should be no spaces between the certificates. Then just save that file with a new filename and ensure it still has a valid certificate extension like cer, crt, pem and so on.

Certificate Based Security

The multitude of different certificate based security types that exist are staggering. But since this document is focused on the 84-Series handset, this section will cover the ones relevant to 84-Series. We'll start with Wi-Fi security that uses certificates in WPA2-Enterprise. Then cover certificates in other connection types for provisioning, CMS integration and PBX integration, among others.

Wi-Fi Security – WPA2-Enterprise

Wi-Fi security with certificates is the exclusive territory of WPA2-Enterprise authentication types. The main one the customers use is EAP-TLS, but you may also use PEAP. How each are used is quite different in 84-Series handsets. We've already talked about PEAP and EAP-TLS a little bit, so we'll focus on how they're actually deployed and how you might use them.

First, we'll cover PEAP since it's likely to be the most common to be deployed as the easier of the two. This is primarily because the handset only needs the Root CA certificate installed. The primary authentication method is via username and password. Many administrators misunderstand that they can use PEAP without validating the certificate. While there are many systems that do allow this, it does mean that the connection isn't actually secured. The 84-Series handsets will not allow you avoid validating the certificate, so you must load the Root CA certificate. As we discussed in the prior section, the chain of authority here is important. That means that the RADIUS system providing authentication needs to send the entire chain of authority so the 84-Series handset can validate the server properly.

With EAP-TLS, the use of certificates is much more important and complicated. This is primarily because EAP-TLS uses client side certificates for the authentication portion instead of the typical username and password. From the server side, EAP-TLS looks an awful lot like PEAP. The RADIUS will need to send the certificate chain to the client for validation just like in PEAP. What's different is that the client will have a certificate that it presents to the RADIUS for validation. A client certificate is created the same way as any other, with a CSR. On the 84-Series handset, you can actually generate a CSR in the handset via the phone's UI. But you can also generate a CSR external to the handset and then load the private key and certificate into the handset via configuration parameters. With EAP-TLS, the username is instead and Identity; and the password is the certificate. The Identity should be the Common Name of the certificate. The 84-Series handsets do have a built-in factory certificate. If you decide to use the built-in certificate, then keep in mind that you will have to download the Spectralink CA certificates[4] and add them to the trust list in your RADIUS. The handset will be presenting its certificate so the RADIUS will need to have the Spectralink certificates to validate the phone in that situation.

Applications and Certificates

The 84-Series handsets has a number of different application types defined that use certificates. At a high-level, the handset can use certificates for applications related to SIP, syslog, LDAP and browser functions. The 84-Series handset has six slots available for application certificates and two for platform certificates. It's important to understand how these can be used and what they cannot be used for. The application slots can be used by the four applications mentioned here. The two platform slots are another matter though and can only be used for wireless authentication, CMS certificate, and XML API integrations. That can be a bit misleading since

[4] http://pki.spectralink.com/aia/Spectralink%20Root%20CA.crt
http://pki.spectralink.com/aia/Spectralink%20Issuing%20CA.crt
http://pki.spectralink.com/aia/Spectralink%20Issuing%20CA%20BLCAI01.crt

CMS and XML API are both somewhat related to browser, which use application slots. But in this case the uses are programmed into the platform slots.

For SIP functions, this could include SRTP or SIPS, a certificate can be configured into an application slot. This is typically done by providing the phone with the PBX certificate, when it is self-signed, or the Root CA when using signed certificates. The phone will use its built-in device certificate for its own encryption unless you choose to load a separate device certificate. There are matching device certificate slots, meaning two platform device slots and six application device slots. As with the running theme we've mentioned multiple times now, certificate chain of authority is necessary for proper function.

For syslog and LDAP, the functions are essentially the same by providing secure communication with system at the far end. The browser function is used by the micro-browser in the handset and can be used for AMiE Advanced integrations as well.

Appendix A

84-Series Protocol and Port List

Name	Transport	Port	Comment
DHCP	TCP/UDP	67, 68	Dynamic Host Control Program
DNS	TCP/UDP	53	Domain Name Lookup
FTP	TCP	20, 21	File transfers
FTPS	TCP	990	Secure file transfers
HTTP	TCP	80	Web Browser communications
HTTPS	TCP	443	Secure Web Browser Connection
NTP, SNTP	UDP	123	Network Time Protocol
ICMP	UDP	7	Ping
PTT	Multicast	5001	Push-To-Talk
OAI	UDP	5456	OAI Communication (Deprecated)
QBC	TCP	14394 (Default)	845x Quick Bar Code Scanner
RTP, SRTP	UDP	16384-32767	Audio ports - Dynamic
SIP Signaling	TCP/UDP/TLS	5060 (Default), 5070 (CS1K v7.5), 5061	SIP Server signaling
Syslog	UDP	514	Syslog UDP data frames
TFTP	UDP	69	File transfers
Web API (XML API)	TCP/UDP	80, 443, 8080	Web API Messaging

Copyright Notice

Notice

Spectralink Corporation has prepared this document for use by Spectralink personnel and customers. The drawings and specifications contained herein are the property of Spectralink and shall be neither reproduced in whole or in part without the prior written approval of Spectralink, nor be implied to grant any license to make, use, or sell equipment manufactured in accordance herewith.

Spectralink reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult Spectralink to determine whether any such changes have been made.

NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING CAPACITY, RESPONSE-TIME PERFORMANCE, SUITABILITY FOR USE, OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY SPECTRALINK FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY OF SPECTRALINK WHATSOEVER.

Warranty

The *Product Warranty and Software License and Warranty* and other support documents are available at http://support.spectralink.com.

US Location	Denmark Location	UK Location
+1 800-775-5330	+45 7560 2850	+44 (0) 13 4420 6591
Spectralink Corporation 2560 55th Street Boulder, CO 80301 USA	Spectralink Europe ApS Bygholm Soepark 21 E Stuen 8700 Horsens Denmark	Spectralink Europe UK Suite B1, The Lightbox Willoughby Road Bracknell, Berkskhire, RG12 8FB United Kingdom
info@spectralink.com	infoemea@spectralink.com	infoemea@spectralink.com