Conditional Access and Security Considerations of Transactional Broadcast Digital Systems

Louis D. Williamson
Time Warner Cable

Abstract

The Orlando Full Service Network (FSN) will be one of the earliest deployments of a true digital interactive broadcast network. Most conditional access systems found in LAN/WAN applications assume an "honest" client. As we move deeper into consumer entertainment, this conventional model demonstrates its limitations. This paper explores the technical and operational considerations for the conditional access system's requirements in an highly interactive environment.

INTRODUCTION

A conditional access system is a security system that will allow a customer to use all services that they have purchased, and deny them access to services they have not purchased. Typical services that are protected in cable television are; premium channels, tiered channels, PPV, and more recently, digital music and Sega channel. Security is the protection of sensitive information that is communicated over the cable system. Types of items that fall under security are; password protection, code download, credit card verification, PIN numbers, e-mail etc. Both conditional access and security are now becoming issues that must be dealt with in cable systems as we began to deploy distributed, transactional networks.

Conditional Access

Presently, conditional access is a function that is handled by converter manufacturers and billing system vendors. When a customer orders a service, a customer service representative (CSR) inputs this information into the billing system. The billing system logs this input into a database and sends an instruction to the converter's headend computer. The headend computer relays this instruction to the converter via an encrypted data stream in the vertical blanking interval of a video channel or in an "out of band" signaling channel. The encrypted instruction tells the converter what channels it can tune and descramble, and what functions it is authorized to perform. PPV is a special case of this authorization process. The difference between PPV and premium channels are that for PPV services most converter are pre-authorized with a defined number of event credits. When a subscriber orders an event, the converter authorizes itself for the event, if the subscriber has any credits. The converter then stores the knowledge that the event has been purchased until it is polled by the headend computer. These traditional types of services only involve the converter manufacturer, the billing vendor, and a CSR when a customer is changing levels of service such as adding or deleting a premium channel.

On an interactive network, conditional access has to be handled by a number of different vendors. If a customer orders a premium channel or a PPV event, this has to be handled through the analog converter's conditional access system. This can be handled the same way it is today. But what about other services such as interactive games and VOD? These services aren't delivered through the analog portion of the set-top, they will be delivered through the digital portion of the set-top. To order a movie, the subscriber needs to get access from the VOD provider. To order a game, one would need to get access from the game vendor. The game may be billed independently from the cable bill. This creates a situation where one may have to get a credit card verification in order to play a game. This transaction would require the subscriber to get

access from the game provider and the game vendor to get credit card verification. Both of these actions require that a secure path be provided from the digital portion of the set-top to the service provider or to the billing system. But neither of the actions listed involves the analog portion of the set-top. Conditional access to these services will have to be controlled by an entirely different method.

As demonstrated above, the billing vendor and converter manufacturer are not the only vendors who are involved in the conditional access function on an interactive network. There will be many other service providers that need to give or deny access to their products. This array of services and different vendors forces one to take a different approach to control access in an interactive network. It also points out that current analog conditional access systems cannot be used to control access to digital services since they aren't involved in the process.

The interactive networks that are envisioned are more like the LAN/WAN environments that exist today. These environments are distributed computing environments. On these networks there are many clients, or customers. There are also many servers, or application providers. These networks have implemented their own type of security. One popular method that is used to secure a LAN or WAN is to use a Kerberos authentication system. This system uses DES encryption in conjunction with user logins and passwords to protect the network.

Unfortunately, it is difficult to force our subscribers to use logins and passwords to gain access to our system. To make this type of a security system viable, the access control needs to be transparent to the users. This forces one to put the login features inside the set-top in a cable environment. But, most LANs and WANs don't use much "physical security". The security algorithms that are in use are software programs that run in memory on the machines. The real security in this type of network is in the secrecy of the user's password. It is assumed that users won't give their passwords to others since they will be held responsible for the other person's actions. This type of protection isn't applicable for a cable system. There will be many services that are available that could be used without authorization in this type of environment. But, without some type of physical security the set-tops that are deployed in a cable system could be cloned and sold to dishonest customers so that they could gain free access to cable services.

The security system that is needed for interactive networks needs to take the features from both of these networks and combine them. The physical security of traditional set-tops is needed to simplify the user interface, and the multiple services to multiple customers paradigm of the distributed computing environment is needed to allow for a rich set of new services.

INTERACTIVE NETWORK SECURITY

On an interactive network there needs to be a secure way for messages and commands to be sent across the network to and from subscribers, operating support systems and service providers. The security system should be transparent to every one operating on the network. To build this type of security system, security measures must be embedded into the interactive network itself. This implies that all of the transactions on the network be secured.

There are some fundamental tasks that a security system must perform on an interactive network. They are:

- Protect privacy of messages
- Provide for conditional access
- Protect against viruses

- Protect against illegitimate kernels and application binaries
- Protect against cloning

FSN Security

Security was embedded into the operating system of FSN. When the FSN set-top is turned on, the set-top only has a small bootstrap loader resident in read only memory. All of the software that runs on the set-top, except the bootstrap loader, is downloaded to the box after it is powered up. This gives one the flexibility to change the operating environment and the user interfaces for the set-top at any given time. Due to the fact that the box is totally downloaded with software, it was necessary to begin implementing security procedures the moment the box is turned on. This is the first step in security in the FSN.

The first task that the bootstrap loader does is to get the boot parameters file for the set-top. The boot parameter files is a data file which is being constantly transmitted over one of the digital channels in the neighborhood. The boot parameter file contains all the information the set-top needs to know regarding how to work on the network. The contents of the boot parameters files are encrypted with one of the bootstrap keys that is stored in the set-top. It is essential that the bootstrap key in the set-top be protected. If the bootstrap key is ever compromised, the set-top can be defeated since this key decrypts all of the other keys that are used in operation. There are several safeguards that are used to protect the bootstrap keys. First, the bootstrap key must be stored in a secured microprocessor inside the set-top. This is to guard against visibility to the keys. Second, to extend the life of the bootstrap keys, they are only used during power up to decode the boot parameters file. Third, the boot parameter files are only broadcast in the neighborhood where the set-top is authorized to be. If the boot parameters keys are ever comprised and the set-top is cloned, the cloned set-tops would only operate in the neighborhood where the original set-top was activated. Fourth, there are several bootstrap keys stored in the set-top's secure microprocessor. If one of the keys is compromised, then the set-top can be instructed to use a different key by changing the bootstrap key reference variable in the boot parameters file.

Once the set-top receives the boot parameters file it performs an integrity check on the file. The integrity check is another important security task. The purpose of the integrity check is to verify that the file that the set-top received was originated from a trusted source. The integrity check performs a secure checksum calculation over the boot parameters file. If the value of the secure checksum calculation matches the value that is stored in the boot parameters file, then it assumes that the boot parameters file is from a trusted source. After this important security step, the set-top has enough information to be able to download the operating software or operating kernel. The contents of the boot parameters file are :

Set-top ID,
Bootstrap key #,
$\%_{(\text{bootstrap key \#})}$Boot channel frequency,
$\%_{(\text{bootstrap key \#})}$Boot kernel name,
$\%_{(\text{bootstrap key \#})}$Kernel integrity checksum,
$\%_{(\text{bootstrap key \#})}$Bootstrap integrity checksum,
$\%_{(\text{bootstrap key \#})}$Runtime key,
$\%_{(\text{bootstrap key \#})}$Other needed data.

Note: %'s implies encrypted with key (bootstrap key #).

The set-top now knows the frequency to tune to download the operating kernel and the name of the file that it should download. The set-top now attempts to download the operating kernel. Once the operating kernel is

downloaded, an integrity check is also performed on it. This is to insure that the operating kernel is legitimate, and not being sent by a pirate. The operating kernel and all application software downloaded by the set-top is sent through an integrity check. The integrity check performs a secure checksum calculation on the downloaded operating kernel just as it did on the boot parameters file. If the calculated value matches the value that is in the boot parameters file, the operating kernel is believed to be legitimate. It is then stored into memory and control is passed to the operating kernel.

After the set-top operating kernel begins to execute, it is in charge of security. Since all applications reside on top of the operating kernel, when a request is made to send a message, the message is automatically secured as part of the operating kernel's secure messaging function. The set-top's operating kernel always talks to the network operating system using secured messages encrypted with the "runtime" key. This is the first level of conditional access in an interactive network. Whenever an application requests network services, the operating kernel must make this request on behalf of the application. This process is called session establishment. To establish a session, the application passes a request to the operating kernel specifying the requirements of the session being established. The requirements includes information such as, who the session is with, the type of bandwidth required in the forward and the reverse directions, etc.. The operating kernel takes this request and then encrypts it with the runtime key that it received in the boot parameters file. The operating kernel then transmits the request to the session manger. Session manger receives this command and passes it to the authentication server. The authentication server is the only element in the network that knows the bootstrap keys for set-tops. The authentication server decrypts the message and passes the message back to session manager with a session key. The session key is the encryption key that will be used between the application server software and the set-tops application software. Session manager takes the information that it received from the authentication server, determines where there is enough bandwidth to support the subscriber's request, and creates a "ticket", (Figure 1). The ticket consists of:
Set-top ID,
$\%_{(\text{runtime key})}$session key,
$\%_{(\text{runtime key})}$forward channel frequency,
$\%_{(\text{runtime key})}$forward bandwidth
$\%_{(\text{runtime key})}$reverse application frequency
$\%_{(\text{runtime key})}$reverse bandwidth
$\%_{(\text{runtime key})}$ticket lifetime,
$\%_{(\text{runtime key})}$other relevant data.

Notice that the ticket has a limited lifetime. If a movie is ordered, the tickets life may be valid for 1.5 times the movie length. This ensures that the ticket cannot be copied and replayed to fool a set-top box. Session manager now takes the ticket, encrypts it with the set-top's runtime key, and transmits it back to the set-top. Since the ticket is encrypted with the set-top's runtime key, only this particular set-top can decode the message. Anyone who is listening in on the RF spectrum would not be able to tell where a particular set-top was told to tune. The ticket is also transmitted to the server using a secure key. This insures that service providers have to go through session manager, and billing, to get to a customer. Until session manger gives the server side of an application a ticket to a set-top, there is no way for a vendor to talk to a set-top. The application vendor must use the ticket's information and encrypt it with the session key before it can be recognized by the set-top.

This is how we perform network conditional access in the FSN, it is a part of the secure messaging function of the security system. Once the set-top and the server receive their tickets, they have enough information that

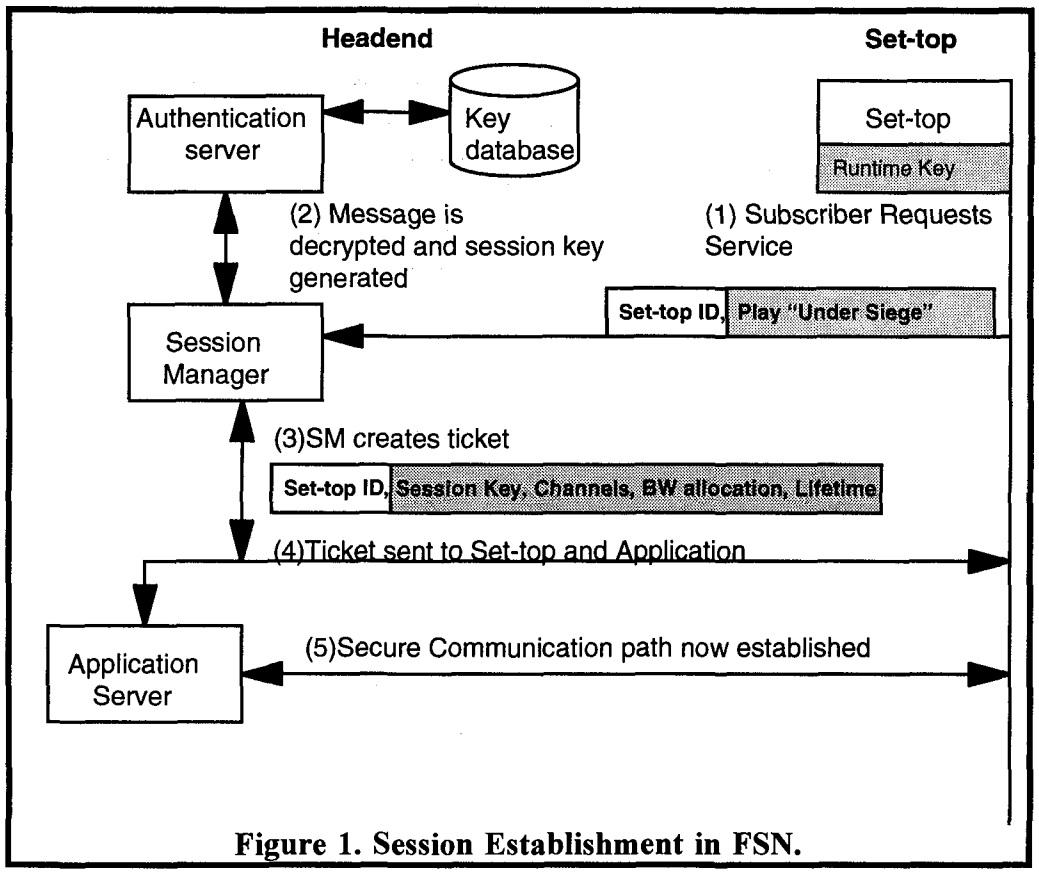

Figure 1. Session Establishment in FSN.

they can communicate with each other. Since the set-top will attempt to decrypt any message from the application provider using the session key, the application provider must use this method if it expects a set-top to understand the message.

Conclusions

The bottom line of any security system is how well it performs over time. This system has yet to be tested on the large scale like converter conditional access systems, but there has been a lot of thought given to how one might break the system. The FSN security does provide for many of the features that are required of security system, but there are some features that are missing.

First, the security provides for secure messaging. All messages are secured when they are transmitted over the network. This protects customers and vendors against snooping. Anyone snooping on the network will have to know the proper key in order to decrypt a message. It gives message privacy, since any application that attempts to talk to a set-top must be given a secure ticket in order to get network services. If an application needs to get credit card information from a customer, then they can be sure that only the application and the customer can decode the message. This also provides for the first level of conditional access since a ticket must be given by session manager before anyone can communicate on the network.

Second, it protects against viruses and bogus codes. Since knowledgeable pirates might try to fool a set-top with a phony applications that could potentially steal credit card information or other private information, the integrity check verifies that all software has come from a trusted source. If someone attempts to load a virus onto the network, this

feature would also prevent it from being loaded and executed.

Third, the security system is seamless to applications and to customers. No one has to think about whether the transaction is at risk. All messages will be encrypted automatically with a session key. Since the session key is random and has a limited lifetime, it is difficult for a pirate to receive information and decrypt the information using powerful computers. By the time the key is discovered, it will probably be invalid.

And finally, the bootstrap key is protected in a secure microprocessor. The bootstrap key also has a limited neighborhood scope. It is only valid in the neighborhood that the set-top is authorized in. This should minimize attempts to clone a set-top since a cloned set-top would only be useful to the subscribers in a 500 home node. The life of the bootstrap key should also be extended since it is only used when the set-top boots. There are also multiple bootstrap keys in the set-top. If one of the keys is compromised another key can be used by simply changing the boot parameters files.

One of the problems with the FSN security is the visibility of the runtime key and the session keys on the computers bus. Even though these keys have limited lifetime to really make this a secure system these keys need to be hidden within the secure microprocessor. The only output of the secure microprocessor should be decrypted messages. Another problem is that currently MPEG video is sent in the clear. There also needs to be a high speed decryption path through the secure microprocessor for the MPEG video data. Since the security process described is software, there isn't enough processor speed to decrypt the video stream and to perform other needed functions. But this task could easily be done in secure hardware, since the video streams are typically running at less that 6 Mbits/Sec.

Will this be enough security to protect the increasing amounts of information that flows across future networks? Only time will tell, but if anyone has begun thinking about potential problems with the FSN security system, and ways to fix the problems then I have succeeded in my task.

REFERENCES

"Base-Level Authentication in the FSN: Overview" ,by Anil R. Gangolli; Silicon Graphics Inc.; Internal FSN document; January 31, 1994.

"Secure Distributed Computing", by Jeffrey I. Schiller; Scientific American; November 1994.