Facsimile Intrusion Systems over IP Networks

A Thesis

Submitted to the Faculty

of

Drexel University

by

Anup Dandekar

in partial fulfillment of the

requirements for the degree

of

Master of Science in Electrical Engineering

September 2008

Acknowledgements

I would like to extend my deepest gratitude to my advisor Dr.Moshe Kam for his guidance and support while pursuing my Masters Degree at Drexel University. It is a honor to be able to work under such a respectable mentor.

I am also grateful to Pramod from the Data Fusion Lab for his genuine support. He has been a amazing colleague and a great friend. Also, I am thankful to the other members of the Data Fusion Laboratory which I was a part of.

Last but not least I would express my thanks to my parents and my brother who have been a continuous source of encouragement in every endeavor of my life.

Dedications

To Mom and Dad

and

my brother Yogesh.

Table of Contents

List of Tables

List of Figures

I did it because I could. – Anonymous

Abstract

Facsimile Intrusion Systems over IP Networks

Anup Dandekar
Advisor: Dr.Moshe Kam

In this thesis, we investigate the security vulnerabilities present in sending a facsimile document over an IP network. We have developed and tested an intrusion mechanism, which is capable of intruding into the facsimile communication over the Public Switched Telephone Network (PSTN) section of the FOIP network without either communicating parties having any knowledge of intrusion taken place. Additionally, we describe the various techniques by which intrusion can take place over an IP network. Finally, we conclude by suggesting countermeasure to prevent the occurrence of such attacks on the FOIP network in the future.

1. Introduction

Security of Facsimile communication over the Internet is a growing concern as the volume of facsimile data sent over the Internet continues to grow. Using the Internet, users are able to transmit facsimile data with reduced cost and less equipment when compared to traditional methods used in the period 1970-1990. However, the Internet is also a potential breeding ground for intruders and attackers who can mount potent attacks on facsimile communications [8]. This environment provides the motivation for the experiments discussed here:

The main concerns with fax messages sent across the network are:

1. Privacy of the message
2. Integrity of the message
3. Authenticity of the message; and
4. Prevention of impersonation and substitution.

1.1 Design Objective:

In this study, we document the various ways in which we can implement attacsks on a FOIP system. Some of these issues are shared with facsimile on telephone lines [9] [11] and some are Internet specific [8].

1.2 Organization of this Document

Chapter 2 provides a list of the terms and protocols which are used in the FOIP system. Chapter 3 provides a review of how a FOIP system works. Chapter 4 discusses the general security concerns which are present in a FOIP system. Chapter 5 provides a means of demonstrating some of these system attacks. Chapter 6 shows a means of implementing a secure FOIP solution. Chapter 7 lists the different cryptographic solutions which can be used to implement a secure FOIP system. We conclude in Chapter 8 with a summary of the work done and proposed work for the future.

2. Terms and Definitions

The availability of IP networks such as the Internet for international communication provides the use of this transmission medium for the transfer of Group 3 facsimile messages between terminals. Since the characteristics of IP networks differ from those provided by the PSTN or ISDN, some additional provisions need to be standardizd to maintain successful facsimile operation.

Facsimile over IP transmission involves a number of protocols working together. Understanding of how these protocols work is crucial in identifying the security threats present in the FOIP system.
Some of the protocols involved in a FOIP transmission are :

- T.38
- T.30
- Session Initiation Protocol (SIP)
- H.323

In this section, we give a description of how these protocols work and are implemented in a FOIP network.

2.1 T.30

ITU-T Recommendation T.30 specifies the procedures for document facsimile transmission over the PSTN [1]. Making a facsimile call and sending a document is divided into five time phases :

1. Phase A, Call Setup : The calling facsimile dials the telephone number of the receiving facsimile machine. The ring signal and the calling tone (CNG) are received at the called facsimile machine. The CNG tone beeps indicate the call is from a facsimile machine instead of a voice call. The called facsimile machine answers the ring signal by going off hook and wakes up the rest of the facsimile machine. The facsimile machine can be designed to have the AC power off until the ring signal arrives. Some facsimile machines answer an incoming call immediately,so the ring may not be heard. After a 1-sec delay, the called facsimile sends a 3-sec, 2100-Hz tone, back to the calling facsimile machine.
2. Phase B, Premessage procedure : The called facsimile machine sends its digital identification signal (DIS) at 300 bps identifying its capabilities, including optional features. On hearing this distinctive

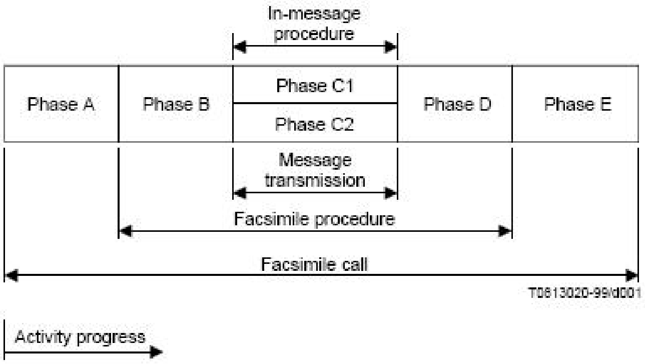

Figure 2.1: Time sequence of a facsimile call,Taken from [1]

signal, the caller presses the send button to connect the facsimile machine to the telephone line (Most modern facsimile machines do this automatically).The calling facsimile automatically sends a digital command signal (DCS), locking the called unit into the capabilities selected. The calling facsimile sends a high speed training signal for the data modem. The called facsimile sends a confirmation to receive (CFR) signal to confirm that the modem is trained (adjusted for low-error operation) and that the facsimile machine is ready to receive.

3. Phase C, Message transmission : The calling facsimile sends a training signal and then picture signals for the entire page being sent.

4. Phase D, Postmessage procedure : The calling facsimile sends a return to control (RTC) command,switching the facsimile modem back to 300 bps ,then an end of procedure (EOP) signal. The called facsimile sends message confirmation (MCF), indicating the page was received successfully.

5. Phase E, Call release : The calling facsimile sends a disconnect (DCN) signal, and both facsimile machines disconnect from the telephone line.

2.2 T.38

This Recommendation [3] defines the messages and data exchanged between facsimile gateways to be applied to allow Group 3 facsimile transmission between terminals where in addition to the PSTN or ISDN a portion of the transmission path used between terminals includes an IP network, e.g.Internet.

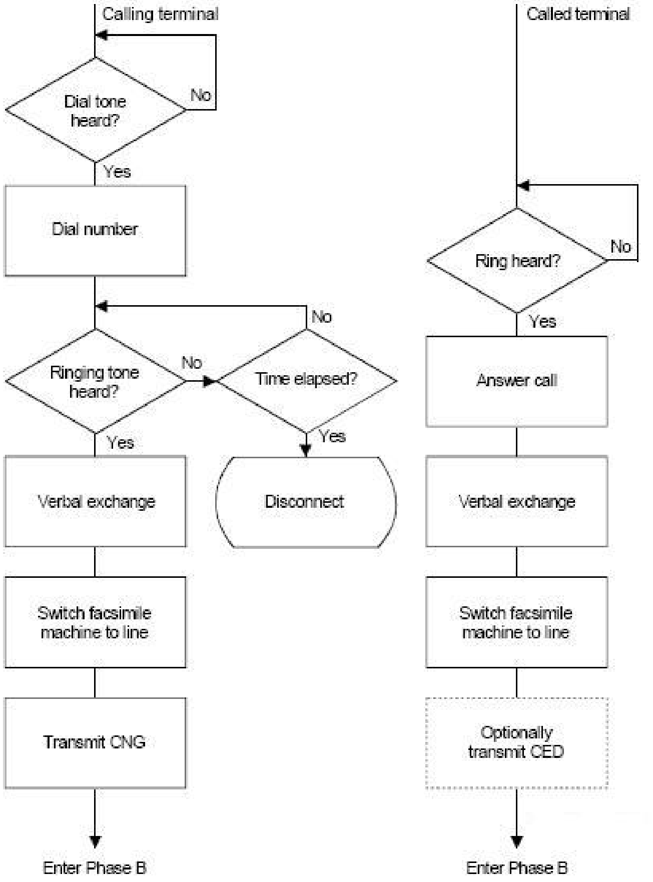

Figure 2.2: Call establishment procedure assuming manual operation at transmitter and receiver,Taken from [1]

2.2.1 Communication between gateways

Internet Protocol-TCP or UDP

The public Internet service provides two principal modes of data transmission :

- TCP(Transmission Control Protocol): A session based,confirmed delivery service;
- UDP(User Datagram Protocol) : Datagram service, non-confirmed delivery.

This recommendation allows the use of either TCP or UDP depending on the service environment.It defines a layered protocol such that the T.38 messages exchanged for TCP and UDP implementations are identical.

Gateway Facsimile data tranfer functions

The emitting gateway shall demodulate the T.30 transmission relieved from the calling terminal. The T.30 facsimile control and image data shall be transferred in an octet stream structure using the IFP packets,over a transport protocol (TCP or UDP). The following signals are not transferred between gateways but are generated or handled locally between the gateway and the G3FE : CNG, CED, and in one mode TCF. The gateways may indicate the detection of the tonal signals, CNG and CED so that the other gateway can generate them.

The receiving gateway shall decode the transferred information and establish communication with the called facsimile terminal using normal T.30 procedures. The receiving gateway shall forward all relevant responses from the called terminal to the emitting gateway.

2.3 SIP-Session initiation protocol

SIP is the call signaling protocol which is used to create, modify or terminate multimedia sessions. These sessions can be established with one or more participants. The transport protocol for SIP can be both UDP or TCP.

The design goals of SIP where scalability ,component reuse and interoperabilty. Scalability stands for two things. Firstly, the scalability of the number of sessions. A user can be involved in an arbitrary number of different sessions at the same time. And secondly SIP was designed to work wide-area from the first day. In other words, users can be located far from one another on the network.

The following subsections introduce the basic parts of SIP, the protocol, the components of a SIP environment and finally a simple call flow.

2.3.1 SIP messages

SIP has a similar pattern to the hyper text transfer protocol (HTTP). It is, as well as HTTP,a request-response protocol.SIP moreover borrows much of the semantics and syntax of HTTP, e.g.the textual message formatting,the usage of headers (which is often identical to HTTP) and multipurpose Internet mail extension(MIME) support. This message would invite the user Caller to a session suggesting a phone call using audio stream. The first thing to notice is that the entire message consists of plain text. Secondly it can be easily differentiated between the header part of the message and the body part. The header contains a description of the message type (INVITE),the protocol type(SIP/2.0), the endpoints of the connection (From-field and T0-field) and the type of the message body. The message body describes the session that the caller wants to establish. There is a particular protocol for this purpose, the session description protocol.

SIP methods The first word in the first line, INVITE specifies the SIP method that is used. SIP methods are the requests that are sent to the other party of a call session. The addressed party answers to the request with a response. The response messages consist as in HTTP out of three digits with an associated textual phrase.

Addressing SIP addresses look similar to email addresses. A SIP Uniform Resource Locator(URL) can look for instance caller@here.org. The format is always user @host. The user part of the address can be chosen arbitrarily. The host part describes the domain the user belongs to. The advantage of this kind of address is that for locating a user the domain name system (DNS) can be used. However, DNS cannot be used if the only give user information is a telephone number.

SDP-Session Description protocol The type of message body as well as its length is described in the header fields of the request message. The corresponding fields are :Content -type and Content-Length.
One possible message type is a multimedia session. The properties of the session are specified using SDP.

The properties of the multimedia session that SIP covers are :

- Session name and it's purpose
- IP address and port number
- Start and stop line
- Information to receive media
- Information about the bandwidth of the session
- Contact information of the person responsible for the session.

  The session descriptions are sent by the caller in the body of the INVITE request as well as by the called party in the corresponding response. The SDP body describes the capabilities of both parties the caller and the callee.

2.3.2 Components of the SIP environment

An important functionality of SIP is user location. A user shall not be bound to exactly one host. To provide user mobility the user must report his actual location to some kind of server. One can see that there are several entities necessary to fulfill this functionality. A SIP environment can therefore contain the following components:

- User agents: They consist of a user agent client that issues requests and a user agent server that responds. In most cases both will be located in one user agent.
- Proxy server: They forward requests to other servers, which can also be proxy servers, or they forward them directly to endpoints.
- Redirect server: They get SIP requests and map the message address to zero or more new addresses. These alternative addresses are returned to the requesting host or server.
- Location server: They map a request's address to the actual host where the user can be reached.
- Registrar : They are servers that accept registration request of user agents. Registrars are often co-located together with proxy servers or redirect servers. The existence of a registrar is not required. However, it can be useful for instance for the mobility of users and for the purpose od user authentication or service subscription. This property can also be used to track users.

2.3.3 Simple call flow

Figure 2.3 shows an architecture for a SIP environment. How the call signaling between two SIP terminals looks like is described in figure. It shows a simplified SIP call with the most important elements.

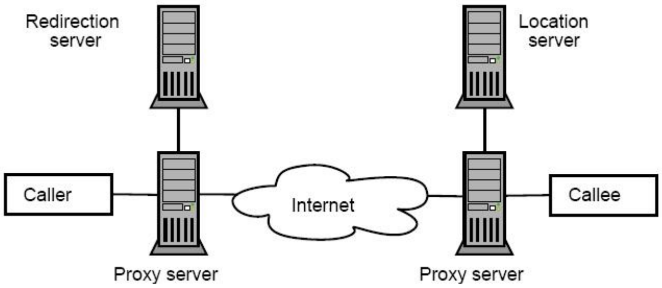

Figure 2.3: A basic call scenario,Taken from [12]

A terminal, phone or endpoint in SIP is called a user agnet. The call starts at the user agent of the caller. The INVITE request is generated and send to the proxy server. The proxy server contacts the redirection server to deliver all the contact addresses for the target user ordered by priority. Upon having received the

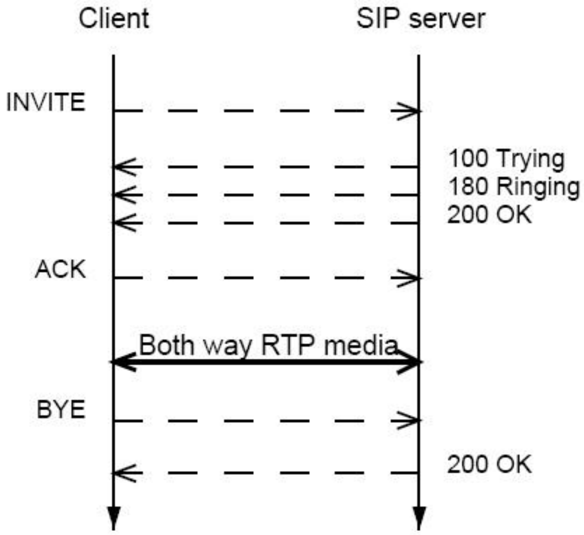

Figure 2.4: Simplified SIP call,Taken from [12]

contact address the proxy server forwards the INVITE to the callee. The callee can also have a proxy server that tracks the location of the user. A proxy server receiving an INVITE request contacts its location server to resolve the location of the callee. It then contacts the user agent in question which responds to the request. The caller send the INVITE request. The user agent of the callee responds with the message 100 Trying indicating that it is processing the call request. The next response is 180 Ringing which signals ,that the user agent tries to reach the user. The callee can either accept or reject the call. The response message for an accepted call is 200 OK. The caller receives the response and sends an ACK message back. With the receipt of that message the callee has also a confirmation about the end of the call initiation. From then on both parties can send their voice traffic to one other. The negotiation of the sessions parameters is done in the INVITE respectively the 200 OK message. For changing the properties of the session either party can send a new INVITE request with the new session description.

The request message and all the corresponding answers build a so called SIP transaction. A session is terminated by using the BYE request. It is acknowledged by the other party by a 200 OK message.

2.4 H.323

H.323 is another protocol which is used to establish calls over the internet. H.323 is a standard of the ITU-T.Recommendation H.323 describes terminals and other entities that provide multimedia services over a packet based network (PBN), which may not provide a guaranteed Quality of service. The support for audio services is compulsory whereas data and video services are optional. For supplementary services like Call Transfer or Call Hold there is another series of protocols, H.450.1-H.450.12.

H.323 contains three main types of interactions. The first of them is responsible for all administrative operations of a user.It is called registration admission status (RAS). The second one takes care of call signalling for call establishment and call completion. This is protocol Q.931.RAS and Q.931 are subsumed to one protocol with the name H.225.0. The third type of functions included in H.323 addresses terminal capability negotiation and call control. These are covered in H.245.

This section starts with introducing the components of an H.323 IP telephony system, and later describes the call set up procedure.

2.4.1 H.323 zones

A zone is the collection of all H.323 entities managed by a single GK. It can also be distributed over several network segments. These segments are connected through routers. There are different types of entities in an H.323 based IP telephony network. They are classified according to their individual function:

- Endpoint(EP) : They are the equivalent to telephones. They can initiate and answer calls. Furthermore they are bale to create and to terminate the data streams necessary to communicate.

- Gateways : Its purpose is the connection from an IP Telephony system to the conventional telephony system (Public switched telephone network(PSTN)). They translate the signaling from from PSTN into H.323.

- Gatekeeper (GK) : A GK serves as an administration unit. It provides five different types of services:

- Multipoint Control Unit(MCU) : It offers the capability for more than two entities to communicate with one anther. It consists of one multipoint controller (MC) and optionally one or more multipoint processors (MP). The MC provides the basic functionality for multipoint conferencing. This includes ad-hoc multipoint conferences as well as capability exchange for all entities involved.Ad-hoc multpoint conferences are calls between two entities that are enlarged to a multi point conference through the participation of another party.A multi point processor is responsible for the central processing of audio and, video data of multi point conferences.

- Administrative Domain Back End Service : The AD-BES or just BES is the service interface for EPs and GK. It stores data about all endpoint types and their permissions, services and configuration. There are two ways to implement the BES. The first is that the EP communicates directly with the BES. In this case, the messages intended for the BES are forwarded by the GK to the BES. In the second case, the EP only interacts with the GK. The GK exchanges messages with the BES to get all the necessary information to respond to the EP requests.

2.4.2 Call Flow

From an architectural point of view there are two possibilities of call establishment, EP-routed call and GK-routed call. In the EP-routed call model an EP directly connects to the desired party. In this case all the traffic is exchanged directly between the two EPs. This includes all protocols of H.323 (H.225 and H.245) and implies that the EP setting up the call exactly knows where the destination resides. This is to say, it has to know the destination's IP address.

In the GK-routed call model, all calls are established through a GK. This model is scalable and administrable and therefore is used in most business models, e.g a subscription based service. Here, every EP has a shared secret. This shared secret is used to prove the identity of the EP's user to the GK. Without this authentication a user is not allowed to establish a call, not even to register. All considerations in the following chapters will refer to this model.

Media traffic in never routed through the GK. The EPs directly exchange all the media data belonging to their call. The interactions between the EP and the GK can be divided into two categories: the registration process and the call establishment. Both are briefly discussed below.

Registration The system is based on the GK-routed model. In this model an EP establishes calls over a GK. This requires a registration procedure. The registration process is composed of two phases. The GK discovery and the registration process. The messages that are involved in the registration process are illustrated in figure Before an EP can start the registration process it has to discover its GK. The first message an EP issues is gatekeeper request(GRQ). If the EP allowed to register with the GK, the GK returns a gatekeeper confirm (GCF) message. The answer in case of a failure is gatekeeper reject(GRJ). A GK discovery is always carried out when an EP is not registered at a GK. This can for instance be either on start up or when a client loses its connection to the GK.

The actual registration is initiated by the EP. The message used for this purpose is registration request(RRQ). The answer of the GK is registration confirm (RCF) in case of success and registration reject(RRJ) otherwise. To unregister an EP sends an unregistration request(URQ). This message does not have to be answered by the GK. In figure, the GK sends a response. A different situation is the deregistration by the GK. In this case the GK sends the unregistration request (URQ). Unlike the deregistration of the EP the answering party must send a response.

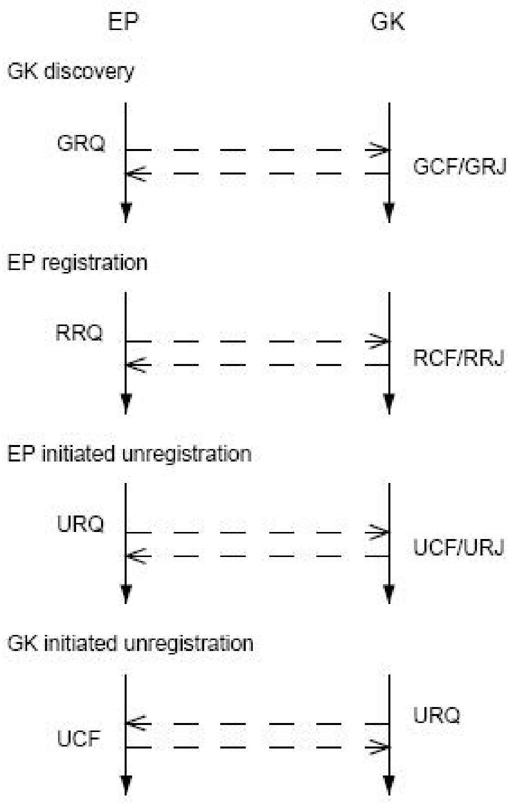

Figure 2.5: Messages involved in the registration process,Taken from [3]

Call establishment This procedure includes RAS messages and the Q.931 protocol. Figure shows a call setup over two GKs. The first step in a call is to request admission. The corresponding message is the admission request(ARQ). If the GK permits the call by sending an admission confirm (ACF) message, the real call establishment can take place.

The next step for the EP is to issue a setup that is forwarded to the destination. The destination EP also has to request admission to answer the call. The following response is connect message. This message is the last one of the call establishment H.225.

After that the terminals have to go through the control procedure H.245 for exchanging the EPs' capabilities and for opening channels for the transport of RTP media.

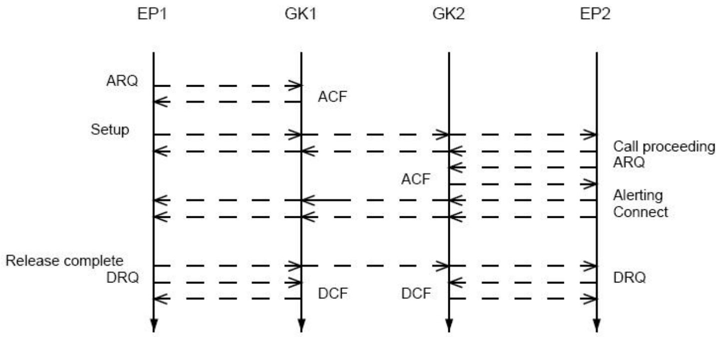

Figure 2.6: Messages involved in the call establishment, Taken from [3]

Call termination The first and main message in the call termination process, as figure illustrates,is the Q.931 message release complete.This message indicates that an ongoing call shall be terminated. An EP issues it and the GK forwards it to other party of the call.The other terminal does not have to return a response (DRQ). The DRQ message is sent to the GK to inform it about the end of the call. The GK sends back a disengage confirm (DCF). Sent by the GK, a DRQ forces the EP to drop a call. The EP is not allowed to reject this request. Thus, it may not send a DRJ received from a GK.

3. How Fax Over IP (FOIP) Works

This chapter describes the procedure in which a fax document is transmitted across a IP network. The procedure is explained with respect to figure 3.1.

The standard components of a FOIP solution include fax terminals and gateways. A fax terminal may be fax equipment designed to work on the PSTN or instead it may be software hosted on a computer. In the former case, a gateway is used to convert the T.30 protocol signals [1] carried on the PSTN into either T.37 or T.38 [2] [3] Internet fax packets for the IP network. If the fax terminal is an IP network attached device (such as a fax application on a computer), then it may contain a virtual gateway.

The ITU-T Recommendation T.30 defines procedures for the transmission of Group-3 fax over the PSTN. This includes the end to end capabilities negotiations between fax terminals as well as the sending of encoded images. A number of modulation types are used for modulating the T.30 signals, and the transmission of Group 3 fax images is specified in Recommendation T.4.

In order for fax signals to traverse an IP network, gateways are employed to convert between T.30 and T.37 or T.38 protocols for non real time and real time fax respectively. Recommendation T.37 deals with transmission of fax in a store-and forward or non real time manner,usually using email capabilities. Recommendation T.38 specifies packet format for messages and data exchanged between T.38 gateways on IP networks in real time. It also specifies the exchange of messages and data to other Internet Aware Fax (IAF) devices .

The FOIP data processing consist of the following four steps:signaling,encoding,transport, and gateway control:

1. Signaling : The purpose of the signaling protocol is to create and manage connections or calls between endpoints. H.323 and the session initiation protocol(SIP)are two widely used signaling standards for call setup and management.
2. Encoding and Transport : Once a connection is setup, fax must be transmitted by segmenting the fax signal into a stream of packets. The first step in this process is to use an analog to digital converter.

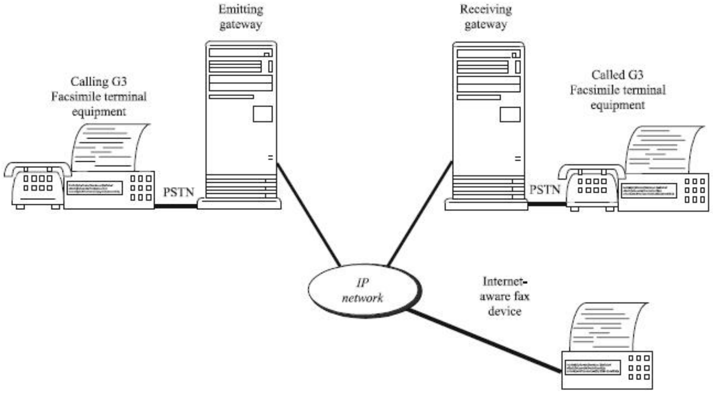
Emitting gateway
Receiving gateway
Calling G3 Facsimile terminal equipment
Called G3 Facsimile terminal equipment
PSTN
PSTN
IP network
Internet-aware fax device

Figure 3.1: Facsimile over IP (FOIP) Network Configuration, Taken from [3]

Here a compression algorithm can be used to reduce the volume of data be transmitted. Next, fax data is inserted into data packets to be carried on the Internet using typically the real-time transport protocol(RTP). RTP packets have header fields that hold data needed to correctly reassemble the packets into a fax signal on the other end. Lastly, the encapsulated fax packets are carried as payload by the user datagram protocol(UDP) for ordinary data transmission. At the other end, the process is reversed : the packets are dissembled and put into analog signals for the called party's machine.

3. Gateway Control : The IP Network itself must then ensure that the real time communication is transported across the telephony system to be converted by a gateway to another format-either for interoperation with a different IP-based multimedia scheme or because the call is being placed onto the PSTN.

With the switch to the Internet as a carrier for fax traffic, we see some of the same security issues that are prevalent in the circuit switched telephone network such as eavesdropping and toll fraud.

4. General Security Concerns in FOIP

This chapter gives a general introduction to system security with a focus on network security. It lists the different types of threats and attacks, as they appear on the Internet.

Each of the components in a FOIP system offers potential vulnerabilities for attackers to exploit .Many of the vulnerabilities are similar to those of the public switched telephone system. For example in the eavesdropping can be achieved by physically placing a listening device on the phone line.
A number of vulnerabilities in the FOIP communications are explored [as given in [6] which are intended for VOIP but can be easily extended to FOIP]. Many of these vulnerabilities can be exploited using a variety of attack vectors. For, many of these attacks there exist counter measures that are available or are being developed.

4.1 General Network Attack Methods

1. Eavesdropping Confidentiality:

   Eavesdropping is when a victim's conversation is secretly monitored by the attacker. This typically involves receiving the fax data from both parties in the call. This data is then used to use the contents for illicit purposes. Reasonable security is expected of the phone system and the conversations that it carries.

2. Alteration of Voice stream-Confidentiality and Integrity:

   This is a substitution attack or man in the middle attack. The attacker is able to listen to the conversation between the two victims and also alter the conversation. This includes play back of previously captured speech so that the receiver hears a different message that the sender sent. This also has value to an attacker when used with interactive voice response phone systems. After capturing a victim's financial passwords, and depleting funds from an account, the substitution attack could be used when the victim calls the interactive voice response to check balances. The attacker could playback a previous balance to give the victim the impressions that no funds had been removed from an account.

3. Redirection of Call-Integrity and Confidentiality:

   One of the features of the FOIP system is the ability to have a single phone number redirected to wherever is present. It is an advanced feature that gives the caller an easy way to find the person they are looking for by dialing a single phone number. This rich feature becomes a potential risk if the redirection feature becomes compromised by an attacker. The attacker can then redirect the victim's phone number to a location of their choice, thus potentially being able to impersonate the victim by having their calls redirected to the attackers telephone. Other than the simple call forwarding, this type of feature has not been available in the traditional phone system. Numerous new features available in FOIP systems will provide a rich set of tools for the end users to gain productivity and also for attackers to exploit.

4. Caller Identification Impersonation-Integrity:

   Each phone has an identity (phone number) that is associated with the device. Having a device impersonate the identity of another can be used as an attack to either receive calls or place calls with the spoofed identity. An attacker wishing to impersonate someone would setup their FOIP device to use the identity of the victims device. The attacker device then registers with the phone system. Any calls intended for the victim's phone number would then be directed towards the attacker's phone. The attacker could then answer a call and impersonate the victim. The attacker also has the ability to use the spoofed device to place calls. The caller id at the phone receiving the call would indicate that the caller is calling the victim, not the attacker.

4.2 Attack Vectors in FOIP and Critical Challenges:

The threats to the FOIP System can be further broken down into specific attack vectors to disrupt the system. For example, an attack mechanism exploiting the OS vulnerabilities can happen at the terminal, server, or other network locations. However, this type of attack all happens at the application layer.
We will focus on the attacks that are specific to FOIP applications and critical to FOIP security assurance :

1. SIP Registration Hacking:

   The SIP is an application layer control protocol that can establish, modify, or terminate user sessions. In SIP and other FOIP protocols , a user agent phone must register itself with an SIP proxy/registrar(control node), which allows the proxy to redirect calls to the phone. Registration Hacking occurs when an attacker impersonates a valid user to a registrar and replaces the legitimate registration with its own address. This attack causes inbound calls intended for the user to be sent to the rogue user.

   Registration hijacking can result in loss of calls to a targeted user. This may be an individual user, group of users, or a high traffic source, such as a media gateway, all inbound calls can be blocked or otherwise manipulated. A rogue user acting in the middle can also record call contents.

Currently, UDP and TCP are used to carry the registration information between the user and the control mode. Utilizing transport layer security(TLS) to create an authenticated secure connection in place of the open connection will prevent SIP registration hacking.

2. SIP Message Modification:

   SIP message have no built in integrity mechanism. By executing one of the man in the middle attacks (IP spoofing, MAC spoofing, SIP registration), an attacker can intercept and modify an SIP message, changing some or all of the attributes of the message.

   This could include the person being called in a session initiation message, giving the victim the impression that he was calling one person while the system connects them to another. By modifying the SIP message, the attacker could impersonate a caller or reroute a call to an unintended party.

   By protecting UDP and TCP transport mechanisms with TLS, the contents of the SIP message are protected. This would prevent an attacker from both reading an SIP message and being able to know who was entering into a call as well as modifying the message to create call fraud.

3. SIP Cancel/Bye Attack:

   The attacker can create an SIP message with the Cancel or Bye Command in its payload and send it to an endnode (phone) to terminate an ongoing conversation. If the attacker sends a steady stream of these packets to the phone, the phone will not be able to place or receive calls. This could be expanded to numerous phones create a system wide disruption of service.

   The lack of strong authentication in SIP makes this attack possible. By adding strong authentication to the communication between the user and the control node, this type of attack can be prevented. The user would verify that the incoming Cancel or Bye command was coming from a trusted node using certificate based credentials.

4. Malformed SIP Command:

   The SIP protocol relies upon an hypertext markup language (HTML) like body to carry command information. This makes the SIP very flexible and extensible for implementing FOIP features. The downside is that it becomes very difficult to test the SIP parser with every possible input. Attackers can exploit these vulnerabilities as they find them by forming packets with malformed commands and sending them to susceptible nodes. This will either degrade or decommission the node that is attacked making part or all of the VOIP system unavailable.

   This is difficult to fix from the perspective of error proofing the message parser. The numerous bugs and their resulting exploits seen in Internet browsers show us how difficult it is to test every possible message that may be sent. Testing with both a dictionary of test cases as well as testing with fuzzing should be performed. Fuzzing software provides intelligently malformed requests that attempt to drive out bugs in software that parses HTML like grammars.

Adding strong authentication to the FOIP SIP command system will also help by preventing an attacker from being able to send malformed SIP commands to a node. This then leaves the authentication protocol as the potential attack point. If there are weaknesses in the parsing logic for the authentication, this attack may still be successful.

5. SIP Redirect:

   SIP employs a server application that receives requests from a phone or proxy and returns a redirection response indicating where the request should be retried. This allows a person to have a call made to him ring at a different phone depending on where they are located, but the caller only dials a single number reach the person. By attacking the redirect server and commanding it to redirect the victim's calls to a number specified by the attacker, the attacker can receive all calls intended of the victim. If the attackers wish to disable the phone network, they could redirect all users' phone numbers to a nonexistent or null type of device. All calls within the FOIP system would then be routed to this nonexistent extension effectively disabling the phone system from delivering calls.

   Once again, this vulnerability is built upon the lack of strong authentication in the SIP Protocol. Moving to a more robust authentication system such as TLS with strong passwords will protect against attacks such as the SIP redirect.

6. RTP Payload:

   The RTP Payload carries the actual encoded fax message between the two callers. It is a simple extension of the UDP protocol adding sequencing information. Using a man in the middle attack to gain access to the RTP media stream between two nodes, an attacker can inspect or modify the payload of the message.

   Inspection in this case becomes eavesdropping on the conversation. If attackers can modify the payload of these messages, they can either inject noise or their own message into the packet. This would either degrade or make impossible conversation between the parties on the call.

   By utilizing Secure RTP (SRTP) protocol, this type of attack can be prevented. The RTP packets are encrypted by the sender and travel the entire network encrypted until being decrypted by the receiver. The SRTP approach prevents eavesdropping and modification of packets to contain new messages.

7. RTP Tampering:

   By manipulation of the sequence number and timestamp fields in the header of the RTP packet, the packets can either be resequenced or made unusable. This attack can either make the conversation unintelligible, or in some implementations of the protocol stack, actually crash the node receiving the packets, thus taking the node offline until the software is reset.

   The SRTP Protocol will allow the receiving node to determine that the RTP header has been modified. This will prevent unusual behaviour from the application software as the packet will be discarded prior to being processed.

   Maintaining the VOIP/FOIP traffic on a local area network (LAN)/virtual LAN separate from the non-VOIP/FOIP traffic will help prevent this type of attack from occuring. Seperating the VOIP/FOIP traffic from the data traffic makes it increasingly difficult to obtain access to the VOIP/FOIP traffic, and thus monitoring or modifying the traffic becomes more difficult.

5. Demonstration of Attacks in an FOIP Environment

In this chapter we demonstrate physical attacks carried out on a FOIP system by implementing a man in the middle attack (mitM) for intercepting a FOIP communication.The location of these attacks are on the PSTN (Public Switched Telephone Network) section of the FOIP system.

5.1 Facsimile Intrusion over PSTN (Method 1):

The first scenario is the MitM attack in which the attacker manages to place itself between the user (host) and the gateway. We devised such a mechanism to intrude at will into an ongoing fax communication and manipulate the fax data. Unless additional measures are taken by the end users, our intrusion is made without the knowledge of either sending or receiving parties. We constructed (Figure 5.1) an ”electronic black box,” which performs the required operation. With the help of this device, we are able to constantly monitor any fax transmissions from the intended sender, and accordingly route the fax message to a third party without the sender gaining knowledge about the intrusion.

The ”black box” we have constructed consists of the following modules:

1. Phone-line simulator: This device simulates a phone line by supplying the required phone line loop current and the dial tone. When a fax machine is plugged into this device, the user will notice no difference with respect to the dial tone and so will be tricked into thinking that he/she is still connected to the actual phone line.
2. Decoder: This unit is used to decode the outgoing phone numbers, which are encoded as DTMF signals.
3. Microcomputer: All the logic is coded into the microcomputer, which controls the switching as well as the decoding and encoding of phone numbers. The software code used in the microcomputer is provided in appendix section.
4. Relay switches: Telecom relay switches are used to switch the black box in and out of the system.
5. Encoder and telephone line loading circuit DAA (Direct Analog Access): The DTMF encoders and the Direct Analog Access circuits are used to send out fax numbers and divert the incoming fax to a third party fax number after intrusion.

The following process describes the intrusion system that we developed:

1. Initially, when the user dials the fax number of the device to which it intends to communicate, DTMF tones are received at the other end of the phone line simulator to which the fax machine is connected.

2. These DTMF tones are decoded by the DTMF decoder circuitry to get the destination fax number.

3. New DTMF tones are generated depending on whether an intrusion is required into the fax conversation. If we wish to intrude into the conversation, we will direct the fax call to us by sending out the appropriate DTMF signals over the phone line or simply regenerate the same DTMF signals so that the fax call is send to the intended recipient (when we do not wish to intrude). These manipulated DTMF tones will be sent over the phone lines with the help of the DAA circuit.

4. After sending out the DTMF tones over the telephone line, with the help of switching relays, one at the telephone line simulator and the other at the other end of the black box, we will pull out of the system in such a way so that the fax transmitter and the corresponding receiver are connected as they were originally for transmitting the fax document. The switching relays are controlled by the microcontroller.

The procedure to implement this as follows:

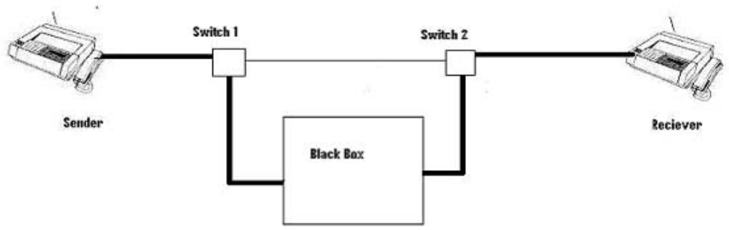

Figure 5.1: Circuit Connections During Intrusion

After the intruding fax machine has received the data, it has to resend the data to the desired receiver, after following a decision as to whether or not to manipulate the data. The operator is informed of the ”opportunity window” to alter the message and resend it and is given the opportunity to upload a new document to replace the original one. The new outgoing document is appended a header that indicates the message came form the original machine.

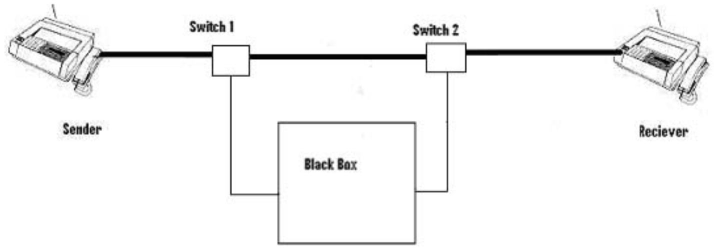

Figure 5.2: Circuit Connections After Intrusion

We were able to demonstrate this process of uploading a document with the header of the original sending machine as well.We programmed the intruding machine with the same details as that of the original sending machine. Next the document was send to the intended receiving machine by appending a new header on the document.

Thus,the whole intrusion process is completed.

5.2 Facsimile Intrusion over PSTN (Method 2):

Another intrusion scenario which we have developed is that of intruding into the fax communication at the receiving end.For this we have used a device placed in the receiver's phone line which basically detects whether the receiver is about to receive a fax message (depending upon the CED contents of the received message). If the call is a fax call intended for the desired recipient then the device routs the call to the intruder machine however if it detects that the call is not from a fax machine then the call is forwarded untouched to the desired recipient.

6. Secure FOIP Solution

In this chapter, we document the various ways in which we can provide a secure FOIP solution. Although a complete list of defense vectors against each threat to the FOIP system is not listed here, we focus to explore the key defense areas which are specific to FOIP applications [as in [6]]. Some of the key defense areas are as listed as follows :

1. Seperation of FOIP and Data traffic : Similar to the port authentication,separating fax and data traffic on to different networks is a key enabler to overall security. Seperating the traffic can prevent a number of attacks as easy as entry into a FOIP network. Due to the expense of running two separate networks,this separation is performed using VLAN (Virtual Local Area Network) technology. The switch of the network implement VLANs by only allowing routing between devices on the same VLAN. Some connectivity between the data and facsimile LAN will be required.

2. Configuration Authentication : The FOIP devices need basic configuration information to get into the system. Configuring the devices provides a classic bootstrap problem where obtaining configuration information from an untrusted source can build into further problems. Preconfiguring fax devices with the public key of various configuration servers at the time of manufacture provides a mechanism for authenticating the authentication server. An alternative would be for the installer of the phone to configure the phone with a public key or shared key or device that would be attached to the phone and provide a quick and accurate means of copying the public key into the phone. The obtain the phone's configuration, the phone would make a DHCP (Dynamic Host Configuration Protocol) request.In the reply from the DHCP server,the phone obtains both its IP address and the IP address of its configuration server. The phone would then establish a connection with the configuration server using the TLS.During the TLS (Transport Layer Security) handshake,the authenticity of the server would be established using the public key the phone device contains and the private key contained in the configuration server.If the two were not a pair,the phone would not load configuration information from the server.If the key pair matches,the configuration information would be loaded into the phone using the FTP over the secure TLS Transport.

3. Signaling Authentication: The connection between the devices in the FOIP network and the servers utilizes the SIP protocol. When a phone registers wit the SIP server,the phone provides an identity. This identity is based upon a number of identifiers including MAC and IP addresses of the phone.Safeguards

are put into place to help minimize the ability of an attacker to spoof the MAC or IP address of the phone. The IP security (IPSec) protocol provides mechanisms for both authentication and encryption. Utilizing IPSec between the VOIP phone and the call manager server provides a strong authentication mechanism.Key sharing provides a strong the basis for the trust between the phone and the server. Key sharing has always been a difficult to solve for large scale deployment.

IPSec provides three different mechanisms for establishing keys. Manual entry into both hosts is the hosts is the least desirable. Also a suitable choice is to utilize the DNS secure (DNSSec) protocol.

4. Media Encryption: Protecting the contents of the facsimile conversation from eavesdropping is a basic concern for many customers utilizing FOIP to conduct business. An extension to the RTP protocol called the Secure real-time protocol(SRTP) has been published by the internet Engineering Task Force (IETF) as RFC 3711. SRTP provides both authentication and confidentiality services for the payload being carried by the RTP protocol. The SRTP protocol has been designed to add a low overhead to the packet size and to minimize the number of key pairs that must be shared between the two communicating nodes. Even with this minimization of key pairs, a single master key pair must exist between the two nodes wishing to communicate. The RFC relies on emerging key exchange protocols for key exchange such as multimedia Internet Keying protocol (MIKEY). MIKEY provides for a lightweight (low bandwidth,low computational needs) protocol capable of exchanging keys in an ad-hoc environment like FOIP calls. Preshared secret keys provide one mechanism for MIKEY to generate session keys. The session keys are used to encrypt the messages sent via the SRTP protocol. A second method of generating session keys depends relies upon public key cryptography. The nodes utilize asymmetric cryptography techniques to generate and exchange a session key. The SRTP session then proceeds just as with the preshared secret keys and utilizes the agreed key to encrypt the SRTP messages uing secret key cryptography. Secret key cryptographic algorithms are always utilized for the SRTP packets as they have a lower compute cost and thus impose less latency on the real time delivery of the voice data. Also, IPSec has been well established and contains a suite of protocols and key exchange algorithms. The Internet Key exchange (IKE) protocol provides a mechanism for two nodes to exchange keys. Using other protocols within the IPSec Suite, a secure tunnel can be established between the two nodes entering into a communication. IPSec works well in establishing secure tunnels as a trunk between organizations. This allows two secure facilities to be connected via a secure link that runs through a nonsecure network such as the Internet. This approach is frequently used to connect branch offices with a central corporate office.

7. Cryptography

Cryptography provides a critical functionality to implement key services. To be able to apply cryptographic methods it is important to know what purpose they fulfill, their functionality and properties. This chapter gives an overview about basic and widely used cryptographic primitives. These are hash functions, symmetric key cryptography and asymmetric key cryptography.

7.1 Authentication

Authentication is basic in the electronic world today. The need to be able to be sure that the person we are talking to is actually who the person claims to be.

Successful authentication can be described as a protocol carried out by two parties A and B,where A authenticates B and B accepts the identity of A. Reusability means the reuse of information obtained during the authentication to impersonate one party at a third party. In this example a reuse of information would be if B could use the data from the performed identification protocol to authenticate at a third party C as A. An identification mechanism shall also avoid impersonation. In the example used here, this would be an impersonation of A to B. These properties must remain true, even if the adversary is able to observe a large number of authentications between A and B.

The security of authentication can be build on something known like a password or a PIN number. The identification mechanism could be a password scheme. Typically,password schemes are weak authentication methods. In general, identification mechanisms have three properties:

1. Identification or entity authentication: It assures on party of both the identity of a second party and the actual participation of the same party at the communication. Here, only necessary data is transmitted. Entity authentication requires a confirmation of authentication through an identifier obtained in the actual communication.

2. Mutual and unilateral authentication: In the first case both parties authenticate to each other whereas in the second case only one party identifies itself.

3. Data origin authentication: It provides to one party that receive a message the assurance of the identity of the party that originated the message. This method typically does not provide timeliness guarantee. The receiver does not have to be in an active communication with the sender. Data origin authentication

provides prove of the knowledge of the key. It includes by definition also data integrity.

Other terms that concern authentication are: message authentication and transaction authentication. Message authentication is a term used analogously with data origin authentication. It provides data origin authentication with respect to the original message source. Transaction authentication provides message authentication together with uniqueness and timeliness guarantees. The last two properties are typically obtained through time variant parameters. These include random numbers, time stamps and challenge response protocols. Another security service that is often provided together with authentication is data integrity. It assures to a second party that data did not change on the way from its sender to the receiver. The reason that data origin authentication and data integrity often are provided at the same time lies in the cryptographic mechanisms used.

7.2 Hash functions

Hash functions play an important role in modern cryptography. They have three basic properties:

1. Preimage resistance: It means that it is computationally infeasible to find an input which hashes to a certain output.
2. 2nd-preimage resistance: It is computationally infeasible to find any seconds inputs which have the same output as the specified input.
3. Collision resistance: It is computationally infeasible to find two distinct inputs which hash to the same value.

The first property preimage resistance means that it is not possible with considerable effort to retrieve the input of a hash function by only knowing the output. The probablility for a hash function with a n-bit hash value is 1/2n. A typical usage of a hash function is to apply it to a message and send the mesage together with the output of the hash function applied to the message of the destination. The receiver can prove by applying the hash function to the message, whether the messages were modified during the transmission or not.

7.3 Symmetric key cryptography

The main property of symmetric key cryptography is that for encryption and decryption the same key is utilized. The two participating parties have to agree upon the key before the data is encrypted and exchanged.

This can be reached for example through a protocol that allows the establishment of a key on an unsecured channel. Once, both parties know the key, the sender can begin to encrypt the messages and send them to the destination.The receive is capable of decrypting the messages because it also knows the key. A secure cipher is characterized by the fact that all the security lies in the key, not in the algorithm. In that case the knowledge of the algorithm is insignificant. It is important that the key is hard to guess. Therefore,the key space needs to be accordingly large.

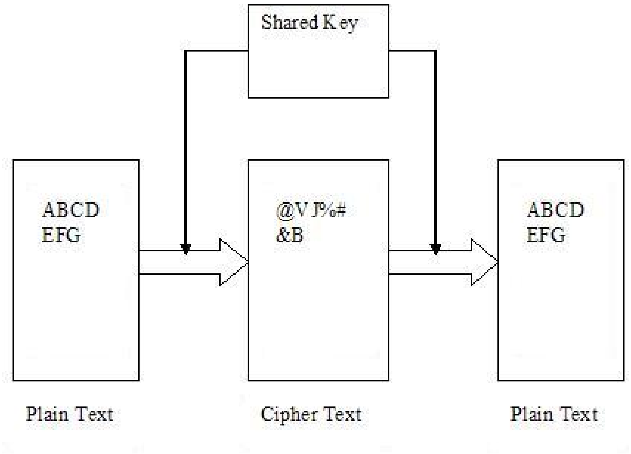

Figure 7.1: Secret Key Cryptography

7.3.1 Stream cipher

Stream ciphers encrypt individual characters (usually binary digits) of a plaintext message one at a time,whereas block ciphers encrypt groups of several characters in one step. Implemented in hardware, stream ciphers tend to be faster than block ciphers. Which one of cipher is more appropriate depends on the kind of application. Stream ciphers are preferred, if buffering is limited or received characters must be processed immediately. Another advantage of stream ciphers is that they have no or limited error propagation which is useful when transmission errors are highly probable.

A stream cipher generates a keystream (a sequence of bits used as a key). Encryption is accomplished by combining the keystream with the plaintext, usually with the bitwise XOR operation. The generation of the keystream can be independent of the plaintext and ciphertext. The result is what is called a synchronous stream cipher. If the key stream depends in some way on the data and its encryption the stream cipher is called self-synchronising. Most stream cipher designs are for synchronous stream ciphers.

7.3.2 Block cipher

Symmetric key block ciphers are the basic building block for a lot of cryptographic mechanisms. They allow the construction of for example hash functions and stream ciphers.
As part of the cryptographic primitives block cipher can serve as a central component in data integrity methods, message authentication techniques, entity authentication protocols and confidentiality mechanisms.

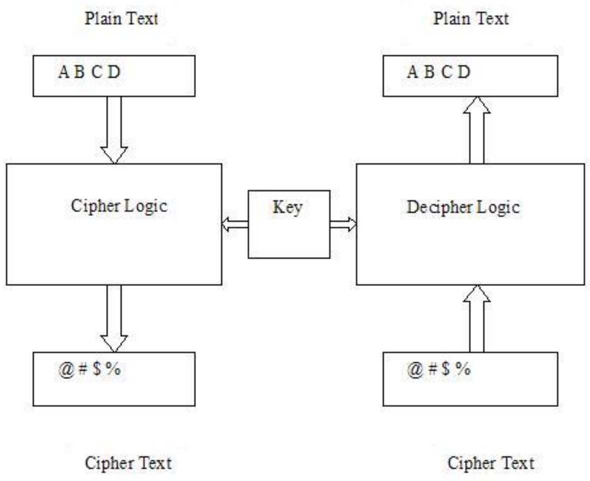

Figure 7.2: A Block Cipher Mechanism

A block cipher is a function that maps plain text blocks with a certain block length to cipher text blocks with the same length. The encryption key is a parameter of this function, the encryption function. To decrypt the cipher text,the encryption function is applied in the reverse direction. The decryption function therefore is the invertible mapping of the encryption function. Furthermore the encryption function must be one to one : each plain text block with a fixed key has to correspond to exactly one cipher text block. The function is

therefore called bijective.
To encrypt messages that exceed this n-bit length there are several so called modes of operation. These specify the way the output and keys of one stage of the algorithm influence the input of the next one. Common modes of operation are : electronic code book (ECB),CBC , cipher feedback (CFB) and output feedback (OFB) [12]. The simplest mode is ECB, where the plain text blocks are enciphered independently from one another.

7.4 Asymmetric cryptography

The primary problem with symmetric ciphers in not their security but the key exchange and the number of keys necessary. If there are n people who need to communicate with one another, there is a need for n(n-1)/2 keys. This is only acceptable for small number of people. Asymmetric cryptography or public key cryptography was introduced to solve these problems. An encryption algorithm based on public key techniques is also called a public key cipher. A public key cipher uses a pair of keys for securely transmitting messages. The two keys belong to the person receiving the message. One key is a public key and may be given to anybody. The other key is a private key and is kept secret by the owner. A sender encrypts a message using the public key. Only the private key can successfully decrypt the message. This approach solves the key exchange problem inherent with symmetric ciphers. There is no need for the sender and receiver to agree upon a key. Before the start of the confidential communication the sender gets a copy of the receiver's public key. Here,only n key pairs are needed for n people to communicate privately with each other.

7.5 Digital signatures

Digital signatures are defined in ISO 7498-2 [4] as: Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of that unit and protect against forgery e.g. by the recipient. Thus, digital signatures are a method to bind the identity of an entity to data. This is an important procedure for authentication, data integrity and non-repudiation.
Signing means generating a tag called a signature, with a secret piece of information that is only known to one entity, the one which signs the data. A signature must be verifiable without the knowledge of the secret key used to create it. The two algorithms that are necessary for generating and verifying the signature are called a digital signature scheme. Generally digital signature schemes can be separated into two classes (both use public key cryptography)[12]: schemes with appendix and schemes with message recovery.
Digital signature schemes with appendix require the original message as input for the verification algorithm. The original message is used to compute a tag that is appended to the message. The tag is only needed to

verify the authenticity of the message. The process of signing consists of two steps. In the first step a hash function is applied to the message. A hash function is used because the generation of a signature is relatively slow in many schemes. The hash function reduces the message to a binary string of fixed length. In the second step the hash value of the message is signed with the secret key.

In the verifying process the verifying entity uses the public key of the signing entity, applies it to the signature and compares the result with the hash value of the message in question. Digital signature schemes with appendix are the most common ones, in practice. These are typically applied to messages with arbitrary length.

In the digital signature schemes with message recovery the original message can be recovered from the signature. Its knowledge is not required to verify the signature. In the signing process, one entity applies a publicly known redundancy function to the message and encrypts the result with its private key. The encrypted part is then the signature. The choice of the redundancy function is crucial to the security of the signature scheme.

The verifying part has to obtain the signature and the public key of the signing party. It then decrypts the signature with the public key and verifies the result. These schemes are in most cases used for messages with fixed length.

In order to use a digital signature scheme in practice one needs a digital signature process. A digital signature process consists of the digital signature schemes well as a method for formatting data into messages that can be signed.

8. Summary and Future Work

8.1 Summary

In summary, we enlist the comprehensive vulnerabilities in sending a facsimile document over a IP network. Many of the vulnerabilities are common to the Public Switched Telephone Network (PSTN). We are able to demonstrate some of the attacks on the FOIP system by physically intruding into the PSTN section of the FOIP system assuming that no precautions are taken at the sending and receiving end of communication for intrusion prevention.

Also,we have mentioned the attacks which are common to any IP network. We have focussed on the attacks which are specific to the protocols involved in FOIP communication.

We conclude by specifying the countermeasures necessary to prevent the occurrence of such attacks on the FOIP network in the future.

8.2 Future Work

An additional form of defense mechanism in the form of intrusion detection [5] [7] needs to be investigated for the FOIP network. To implement a intrusion detecion system for a FOIP network the following points are of importance :

- The special attacks on the FOIP protocols need to be considered while implementing a intrusion detection system for a FOIP network.
- Intrusion Detection will provide an important line of defense to complement intrusion prevention.
- Numerous Intrusion Detection systems have been investigated for other data networks.These can be easily transferred to the FOIP network.

Bibliography

[1] *Procedures for document facsimile transmission in the general switched telephone network*. ITU-T Recommendation T.30, July,1996. Terminal Equipments and protocols for telematic services.

[2] *Procedures for transfer of facsimile data via store and forward on the internet*. ITU-T Recommendation T.37, July,1996. Terminal Equipments and protocols for telematic services.

[3] *Procedures for Real Time Group 3 facsimile communication over IP networks*. ITU-T Recommendation T.38, July,1998. Terminal Equipments and protocols for telematic services.

[4] OSI-Reference Model: Part 2: Security architecture,first edition. Technical report, International Organisation for Standardization, June 2000. ISO/IEC 17799.

[5] Stefan Axelsson. Intrusion Detection Systems : A survey and taxonomy. Technical report, Chalmers University of Technology, Goteborg, Sweden, 2000.

[6] David Butcher, Xiangyang Li, and Jinhua Guo. Security challenge and defense in VOIP infrastructures. *IEEE Transactions on Systems ,Man and Cybernetics*, 37:1152–1162, Nov,2007.

[7] Herve Debar, Marc Dacier, and Andreas Wespi. Towards a taxonamy of intrusion detection systems. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 31:805–822, 2000.

[8] Alan Johnston and David Piscitello. *Understanding Voice over IP Security*. Artech House, Dedham,MA, 2006.

[9] L.R.Berke. Method for certiying facsimile communciations over a telephone network. United states patent 5940187, 1999.

[10] Kenneth McConnell, Dennis Bodson, and Stephen Bordon. *FAX: Facsimile Technology and Systems*. Artech House, Dedham,MA, 3 edition, 1999.

[11] Stephen Perschau. Security and facsimile. Technical Report ADA319870, Delta Information Systems Inc, Horsham PA, July 2,2007.

[12] Johann Thalhammer. *Security in VOIP-Telephony Systems*. Master thesis, Graz University of Technology.

Appendix A. Code from Microcontroller

```
/This program was produced by the CodeWizardAVR V1.25.6 Evaluation
Automatic Program Generator  Copyright 1998-2007 Pavel Haiduc, HP
InfoTech s.r.l. http://www.hpinfotech.com

Project : Version : Date    : 8/6/2007 Author  : Freeware, for
evaluation and non-commercial use only Company : Comments:


Chip type           : ATmega128 Program type         : Application
Clock frequency     : 16.000000 MHz Memory model         : Small
External SRAM size  : 0 Data Stack size      : 1024
*****************************************************/

/*This program is used to do the following operations

1) Process the Decoded incoming DTMF tones from the sending fax
machine
2) Ascertain via the decoded tones whether the called number
is of interest
3) If the number is of interest, send new DTMF tones
in order to intrude by switching the relays
4) If not of interest,
simply send the original DTMF tones.
*/

#include <mega128.h> #include<delay.h>

#define RXB8 1 #define TXB8 0 #define UPE 2 #define OVR 3 #define FE
4 #define UDRE 5 #define RXC 7
```

```
#define FRAMING_ERROR (1<<FE) #define PARITY_ERROR (1<<UPE) #define
DATA_OVERRUN (1<<OVR) #define DATA_REGISTER_EMPTY (1<<UDRE) #define
RX_COMPLETE (1<<RXC)

// Get a character from the USART1 Receiver
#pragma used+ char getchar1(void) { char status,data; while (1)
      {
      while (((status=UCSR1A) & RX_COMPLETE)==0);
      data=UDR1;
      if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
         return data;
      };
} #pragma used-

// Write a character to the USART1 Transmitter
#pragma used+ void putchar1(char c) { while ((UCSR1A &
DATA_REGISTER_EMPTY)==0); UDR1=c; } #pragma used-

// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here
  char k[10];
  int i;
void main(void) {
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTA=0x00; DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00; DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00; DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00; DDRD=0x00;

// Port E initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTE=0x00; DDRE=0x30;

// Port F initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTF=0x00; DDRF=0x00;

// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00; DDRG=0x00;
```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
ASSR=0x00; TCCR0=0x00; TCNT0=0x00; OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// OC1C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0x00; TCCR1B=0x00; TCNT1H=0x00; TCNT1L=0x00; ICR1H=0x00;
ICR1L=0x00; OCR1AH=0x00; OCR1AL=0x00; OCR1BH=0x00; OCR1BL=0x00;
OCR1CH=0x00; OCR1CL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
TCCR2=0x00; TCNT2=0x00; OCR2=0x00;
```

```
// Timer/Counter 3 initialization
// Clock source: System Clock
// Clock value: Timer 3 Stopped
// Mode: Normal top=FFFFh
// Noise Canceler: Off
// Input Capture on Falling Edge
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Timer 3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x00; TCCR3B=0x00; TCNT3H=0x00; TCNT3L=0x00; ICR3H=0x00;
ICR3L=0x00; OCR3AH=0x00; OCR3AL=0x00; OCR3BH=0x00; OCR3BL=0x00;
OCR3CH=0x00; OCR3CL=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
// INT3: Off
// INT4: Off
// INT5: Off
// INT6: Off
// INT7: Off
EICRA=0x00; EICRB=0x00; EIMSK=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00; ETIMSK=0x00;
```

```
// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: On
// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud Rate: 1200
UCSR0A=0x00; UCSR0B=0x18; UCSR0C=0x06; UBRR0H=0x03; UBRR0L=0x40;

// USART1 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART1 Receiver: On
// USART1 Transmitter: On
// USART1 Mode: Asynchronous
// USART1 Baud Rate: 1200
UCSR1A=0x00; UCSR1B=0x18; UCSR1C=0x06; UBRR1H=0x03; UBRR1L=0x40;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80; SFIOR=0x00;

  // putchar(’h’);

  // This section is used to fetch the decoded DTMF tones.

   for(i=0;i<4;i++)
   {
        k[i]=getchar();

   }

     for(i=0;i<4;i++)
```

```
{
     putchar1(k[i]);

}


       PORTE.5=0;

// This section checks whether the DTMF tones are of interest
// If of interest, transmit a different set of DTMF tones
// If not of interest, transmit the original DTMF tones

if(k[0]==’1’)
     {
           if (k[1]==’7’)
                 {
                      if(k[2]==’4’)
                      {

                        if(k[3]==’0’)
                        {
                           delay_ms(2000);
                           putchar(’t’);  // to take the phone off hook
                           delay_ms(1000);
                           putchar(’2’);
                           delay_ms(1000);
                           putchar(’3’);
                           delay_ms(1000);
                           putchar(’0’);
                           delay_ms(1000);
                           putchar(’5’);

                           delay_ms(1000);
```

```
   putchar(’\r’); //to transmit

   delay_ms(1000);

   PORTE.4=1;


   delay_ms(100);

   PORTE.5=1;
   putchar1(’H’);
   delay_ms(300);

   delay_ms(1);
   putchar(’H’);



}
else
{
   delay_ms(2000);
   putchar(’t’);
   delay_ms(1000);
   putchar(’1’);
   delay_ms(1000);
   putchar(’7’);
   delay_ms(1000);
   putchar(’4’);
   delay_ms(1000);
   putchar(’0’);
```

```
     delay_ms(1000);
     putchar(’\r’);
     putchar1(’E’);

     delay_ms(1000);
     PORTE.4=1;
     delay_ms(100);
     PORTE.5=1;
     putchar1(’H’);
     delay_ms(300);
     delay_ms(1);
     putchar(’H’);
     delay_ms(10000);
     putchar(’H’);
  }
}
else
{
     delay_ms(2000);
     putchar(’t’);
     delay_ms(1000);
     putchar(’1’);
     delay_ms(1000);
     putchar(’7’);
     delay_ms(1000);
     putchar(’4’);
     delay_ms(1000);
     putchar(’0’);
     delay_ms(1000);
     putchar(’\r’);
     putchar1(’M’);
     delay_ms(1000);
```

```
                PORTE.4=1;
                delay_ms(100);
                PORTE.5=1;
                putchar1(’H’);
                delay_ms(300);
                delay_ms(1);
                putchar(’H’);

                delay_ms(10000);
                putchar(’H’);
           }
      }
else
      {

                putchar(’t’);
                delay_ms(2000);
                putchar(’1’);
                delay_ms(1000);
                putchar(’7’);
                delay_ms(1000);
                putchar(’4’);
                delay_ms(1000);
                putchar(’0’);
                delay_ms(1000);
                putchar(’\r’);
                putchar1(’B’);

                delay_ms(1000);
                PORTE.4=1;
                delay_ms(100);
                PORTE.5=1;
```

```
                          putchar1(’H’);
                          delay_ms(300);
                          delay_ms(1);
                          putchar(’H’);

                          delay_ms(10000);
                          putchar(’H’);
                }
    }
else
    {                     delay_ms(2000);
                          putchar(’t’);
                          delay_ms(1000);
                          putchar(’1’);
                          delay_ms(1000);
                          putchar(’7’);
                          delay_ms(1000);
                          putchar(’4’);
                          delay_ms(1000);
                          putchar(’0’);
                          delay_ms(1000);
                          putchar(’\r’);
                          putchar1(’F’);
                          delay_ms(1000);
                          PORTE.4=1;
                          delay_ms(100);
                          PORTE.5=1;
                          putchar1(’H’);
                          delay_ms(300);
                          delay_ms(1);
                          putchar(’H’);
                          delay_ms(10000);
```

```
                               putchar('H');

        }


while (1)
      {




      };
}
```