and delivery stage of the project. Therefore, SRS and Software Testing are relatively essential to achieve the overall functionality, quality attributes and performance of the project [9].

Therefore, it is highly essential to execute SRS analysis and ensure that it is properly understood by both parties in the view of customer functional requirements and legal requirements.

Legal non-functional requirements (quality factors) as applicable to the given problem need to be identified, a checklist may be documented and prioritizing them may be beneficial. This list needs to be evaluated, dependency relationship among non-functional and functional requirements have to be identified. The check list needed to be redefined and documented.

In the perspective of cyber space issues, non-functional requirements are the legal constraints under which a system operates. As application of the software changes characterization of NFR changes. There exists diversity in characterization of NFR and it is subjected to the application domain.

It is highly essential to formulate the non-functional requirements. The characterizing and formulating NFR in [20] is centered around quality attributes such as maintainability, modularity, etc but omits the legal issues. This work modifies the formulation to include these key legal issues such as IPR, CL, etc and the same is shown in the table below (Table-1).

Legal issues of the cyber space have to be given attention while eliciting, analyzing, documenting and tracking functional, non-functional requirements and domain requirements. Otherwise, the development may have to be compromise with the quality of the product, the cost to develop and enhance it, and the time-to-market of current and future releases. As mentioned in [9], without quality targets to guide the architects and engineers, design choice are random, and it is tough to assess the system during architecture and design reviews and system test.

Table-1: Formulating NFR including Cyber space issues (IPR, CL, etc).

Non-functional requirements	1. What is your definition of non-functional requirements? 2. Does it defines the cyber space legal issues such as IPR, CL, etc., as part of NFR? 3. What are the general relationships between typical quality attributes and features? 4. What are the legal constraints that bind the specific functional requirement? 5. Verify, validate and obtain the formal approval of the legal constraints identified from a legal authority. 6. How can non-functional requirements (attributes) be allocated to development activities?
Requirements traceability	1. What should be traced? 2. How and when to trace it? 3. How can tracing improve software quality?
Structuring the RE process	1. Is requirements formalization necessary to improve software quality? 2. What's the scope of the user? 3. How to deal with requirements overload?

6. IPR & CL BASED SYSTEM DEVELOPMENT LIFE CYCLE (IPRCL-SDLC)

System Development Life Cycle (SDLC) is defined as the process (as a whole) of developing high quality system or software to meet assured requirements. Requirements Engineering, Design, Implementation, Testing and Maintenance are universally known phases of SDLC. In Each SDLC phase, a Validation and Verification (V&V) sub-phase controls the quality of the deliverables. It is highly desired to carry out the verification and validation (V&V) with respect to functional, non-functional and domain requirements mapped to legal issues also. It is highly desirable and essential to analyze the possible internal relationship that may exist between the functional, non-functional and domain requirements. Generally, in non-legal based SDLC V&V is confined to customer requirement functionalities. Though NFR and DR analysis/design is done, it lacks in terms of cyber space issues. This work devices IPR and CL issues based SDLC (IPRCL-SDLC). In literature survey even though patent based software life cycle is documented (Paul Klint 2006, //homepages.cwi.nl ~paulk/patents /isnot/node7.html), it does not address the question like whether the system/software being developed violates the CL.

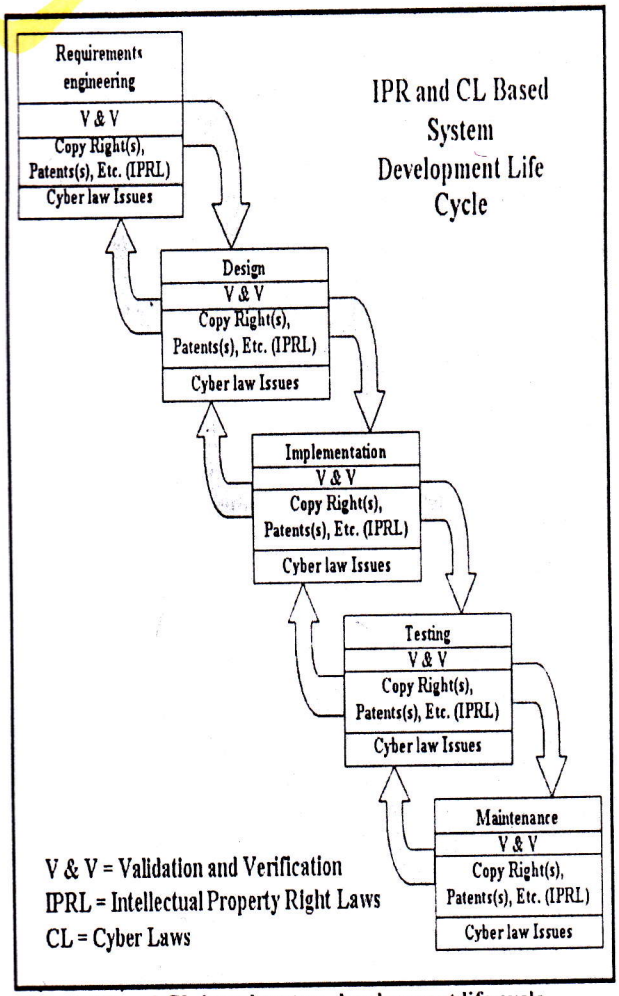


Fig 1. IPR and CL based system development life cycle.

This work has designed IPRCL-SDLC that takes into account IPR as well as CL during software development. In each V&V sub-phase, in addition to questions on Copy rights, patents, etc (IPR), CL issues have to be addressed. The Fig 1, above shows IPR and CL based system development life cycle.

7. CONCLUSION

This work enlightens on the legal issues such as IPR and CL in software Industry and development. It List outs a sample set of research questions to be addressed by industry during the software development. A modified formulation of non-functional requirements including the legal issues is presented. IPR and CL based SDLC brings out the paradigm shift in legal issues analysis and incorporating the same into software development.

In summary, IPR and CL are critical issues in every stage of software development life cycle, right from requirements engineering and failing to incorporate them may lead to cyber accidents, crime, software piracy etc,. These legal issues are inevitable even though it increases the complexity and cost of the software development and to be noted that these are key quality determining factors.

It is needless to say that during the course of software development all models such as models to design, develop, deploy, and manage technology solutions have to be approved by IPR law authorities and CL authorities. Further, quality software product and technologies can be developed if software process engineering is based on Cyber space issues.

To achieve high quality software, requirements engineering (RE) consisting functional requirements, 'non-functional' or 'quality' requirements including cyber space issues have to be elicited from the customer and environment (cyber society) and to be represented in a requirements document in order to provide the software designer and tester a complete and correct specification. Conventional RE methods focuses mainly on functional requirements and to some extent non-functional requirements but Cyber space issues are usually not seen.

In general quality software could be improved by capturing functional requirements, non-functional requirements and applicable cyber space issues. Failing to capture and analyze any one of these issues during requirements engineering does not guarantee the quality. The software engineering is not just applying sound engineering principles that ensure to meet customer requirements but also should focus on legal and ethical issues. The work shows a general formalization (doing complete formalization is difficult and challenging and case specific). The future work includes in depth legal issues based NFR engineering and effective modeling using SoftModToones.

8. ACKNOWLEDGMENTS

ARUN KUMAR B.R, MCA, M.Phil (CS) M.Tech (CS& E) has submitted his Ph.D (CS) thesis to the Dravidian university, Kuppam, A.P, India. The author has published 15 research papers in National/International Journals and 11 papers in the proceedings of National/International Conferences including IEEE international conferences, wishes to place on record his sincere thanks to all those who helped in bringing out this paper.

9. REFERENCES

[1] Dukrok Suh, Junseok Hwang and Donghyun Oh," Do Software Intellectual Property Rights Affect the Performance of Firms? Case Study of South Korea", PP 307-312, 978-0-7695-3372-8/08 ,© 2008 IEEE, DOI 10.1109/ICSEA.2008.73

[2] Rolan Abdukalykov, Ishrar Hussain, Mohamad Kassab, Olga Ormandjieva," Quantifying the Impact of Different Non-Functional Requirements and Problem Domains on Software Effort Estimation", 2011 Ninth International Conference on Software Engineering Research, Management and Applications, 978-0-7695-4490-8/11 © 2011 IEEE, DOI 10.1109/SERA.2011.45, pp-158-166.

[3] Jyh-sheng Ke, Institute for Information Industry, Taiwan," SOFTWARE INDUSTRY IN TAIWAN"

[4] Anargyros Tsadimas, Mara Nikolaidou, Dimosthenis Anagnostopoulos," Handling non-functional requirements in Information System Architecture Design", 2009 Fourth International Conference on Software Engineering Advances, 978-0-7695-3777-1/09 © 2009 IEEE, DOI 10.1109/ICSEA.2009.18

[5] Naavi," Essentials of Cyber Laws for IT Professionals", Presented at Sairam Engineering College on February 18, 2003 during the Seminar on Cryptography.

[6] Randall Davis, Pamela Samuelson,Mitchell Kapor, Jerome Reichman," A New View of Intellectual Property and Software" COMMUNICATIONS OF THE ACM March 1996/Vol. 39, No. 3 page. No. 21-31.

110

FIGURE 13.7
Examples of common hacking tactics to assault companies through the Internet and other networks.

Common Hacking Tactics

Denial of Service This is becoming a common networking prank. By hammering a website's equipment with too many requests for information, an attacker can effectively clog the system, slowing performance or even crashing the site. This method of overloading computers is sometimes used to cover up an attack.

Scans Widespread probes of the Internet to determine types of computers, services, and connections. That way the bad guys can take advantage of weaknesses in a particular make of computer or software program.

Sniffer Programs that covertly search individual packets of data as they pass through the Internet, capturing passwords or the entire contents.

Spoofing Faking an e-mail address or Web page to trick users into passing along critical information like passwords or credit card numbers.

Trojan Horse A program that, unknown to the user, contains instructions that exploit a known vulnerability in some software.

Back Doors In case the original entry point has been detected, having a few hidden ways back makes reentry easy—and difficult to detect.

Malicious Applets Tiny programs, sometimes written in the popular Java computer language, that misuse your computer's resources, modify files on the hard disk, send fake e-mail, or steal passwords.

War Dialing Programs that automatically dial thousands of telephone numbers in search of a way in through a modem connection.

Logic Bombs An instruction in a computer program that triggers a malicious act.

Buffer Overflow A technique for crashing or gaining control of a computer by sending too much data to the buffer in a computer's memory.

Password Crackers Software that can guess passwords.

Social Engineering A tactic used to gain access to computer systems by talking unsuspecting company employees out of valuable information such as passwords.

Dumpster Diving Sifting through a company's garbage to find information to help break into their computers. Sometimes the information is used to make a stab at social engineering more credible.

intruders. A hacker may also use remote services that allow one computer on a network to execute programs on another computer to gain privileged access within a network. Telnet, an Internet tool for interactive use of remote computers, can help hackers discover information to plan other attacks. Hackers have used Telnet to access a computer's e-mail port, for example, to monitor e-mail messages for passwords and other information about privileged user accounts and network resources. These are just some of the typical types of computer crimes that hackers commit on the Internet on a regular basis. That's why Internet security measures like encryption and firewalls, as discussed in the next section, are so vital to the success of electronic commerce and other e-business applications.

The hacking community is quick to make the distinction between hacking and cracking. A cracker (also called a black hat or darkside hacker) is a malicious or criminal hacker. This term is seldom used outside of the security industry and by some modern programmers. The general public use the term *hacker* to refer to the same thing. In computer jargon the meaning of *hacker* can be much more broad. The name comes from the opposite of white hat hackers.

Usually a cracker is a person who maintains knowledge of the vulnerabilities and exploits he or she finds as secret for private advantage, not revealing them either to the general public or to the manufacturer for correction. Many crackers promote individual freedom and accessibility over privacy and security. Crackers may seek to expand holes in systems; any attempts made to patch software are generally to prevent others from also compromising a system they have already obtained secure control over. In the most extreme cases, a cracker may work to cause damage maliciously, and/or make threats to do so for blackmail purposes.

112

FIGURE 13.8
Internet abuses in the workplace.

Internet Abuses	Activity
General e-Mail Abuses	Include spamming, harassments, chain letters, solicitations, spoofing, propagations of viruses/worms, and defamatory statements.
Unauthorized Usage and Access	Sharing of passwords and access into networks without permission.
Copyright Infringement/ Plagiarism	Using illegal or pirated software that costs organizations millions of dollars because of copyright infringements. Copying of websites and copyrighted logos.
Newsgroup Postings	Posting of messages on various non-work-related topics from sex to lawn care advice.
Transmission of Confidential Data	Using the Internet to display or transmit trade secrets.
Pornography	Accessing sexually explicit sites from workplace as well as the display, distribution, and surfing of these offensive sites.
Hacking	Hacking of websites, ranging from denial-of-service attacks to accessing organizational databases.
Non-Work-Related Download/Upload	Propagation of software that ties up office bandwidth. Use of programs that allow the transmission of movies, music, and graphical materials.
Leisure Use of the Internet	Loafing around the Internet, which includes shopping, sending e-cards and personal e-mail, gambling online, chatting, game playing, auctioning, stock trading, and doing other personal activities.
Usage of External ISPs	Using an external ISP to connect to the Internet to avoid detection.
Moonlighting	Using office resources such as networks and computers to organize and conduct personal business (side jobs).

Source: Adapted from Keng Fiona Fui-Hoon Nah and Limei Teng, "Acceptable Internet Use Policy," *Communications of the ACM*, January 2002, p. 76.

Unauthorized Use at Work

The unauthorized use of computer systems and networks can be called *time and resource theft*. A common example is unauthorized use of company-owned computer networks by employees. This may range from doing private consulting or personal finances, or playing video games, to unauthorized use of the Internet on company networks. Network monitoring software, called *sniffers*, is frequently used to monitor network traffic to evaluate network capacity, as well as to reveal evidence of improper use. See Figures 13.8 and 13.9.

According to one survey, 90 percent of U.S. workers admit to surfing recreational sites during office hours, and 84 percent say they send personal e-mail from work. So this kind of activity alone may not get you fired from your job. However, other Internet activities at work can bring instant dismissal. For example, *The New York Times* fired 23 workers because they were distributing racist and sexually offensive jokes on the company's e-mail system [39].

Xerox Corp. fired more than 40 workers for spending up to eight hours a day on pornography sites on the Web. Several employees even downloaded pornographic videos, which took so much network bandwidth that it choked the company network and prevented coworkers from sending or receiving e-mail. Xerox instituted an eight-member SWAT team on computer abuse that uses software to review every website its 40,000 computer users view each day. Other companies clamp down even harder, by installing software like SurfWatch, which enables them to block as well as monitor access to off-limit websites [20].

CSE516 Web Technology

Chapter 6

How do you reduce or eliminate intellectual property theft? How do you ensure message integrity? How do you authenticate a user? What is a firewall and how can it help protect an electronic commerce application?

Implementing Security for Electronic Commerce

Introduction

In this chapter you will learn about the different security measures that can be employed to reduce or eliminate intellectual property theft from electronic commerce applications. You will gain an overview of how to secure client computers from attack by viruses and ill-intentioned programs and scripts downloaded in web pages. By the end of this chapter you should also understand the role secure socket layers, secure http and secure electronic transaction protocols play in protecting electronic commerce websites.

Learning objectives

In this chapter you will learn:

- What security measures can reduce or eliminate intellectual property theft
- How to secure client computers from attack by viruses and by ill-intentioned programs and scripts downloaded in Web pages
- How to authenticate users to servers and authenticate servers
- Protection mechanisms that are available to secure information sent between a client and server so that the information is not disclosed
- How to secure message integrity, preventing another program from altering information as it travels on the internet
- Safeguards that are available so commerce servers can authenticate users
- How firewalls can protect intranets and corporate servers against being attacked through the internet
- What role the Secure Socket Layer, Secure HTTP, and secure electronic transaction protocols play in protecting electronic commerce.

Student reading

It is strongly advised that you now read Chapter 6 of the recommended course text.

'Electronic Commerce' James Perry & Gary Schneider, 2nd ed Paperback, 489 pages Course Technology, 2001 ISBN: 0619033789

Chapter outline

This chapter outline is provided as a guide to your reading for this chapter. If by chance you cannot obtain the recommended course text for this chapter, then you should aim to read texts which cover the following topics:

- Protecting electronic commerce assets

- Protecting intellectual property
- Protecting client computers
  - Monitoring active content
    1. Digital certificates
    2. Microsoft internet explorer
    3. Netscape Navigator
  - Dealing with cookies
  - Using Antivirus software
  - Calling in computer forensic experts
- Protecting electronic commerce channels
  - Providing transaction privacy
    1. Encryption
    2. Encryption algorithms and standards
    3. Secure sockets layer protocol
    4. Secure HTTP (S-HTTP) protocol
- Ensuring transaction integrity
  - Guaranteeing transaction delivery
- Protecting the commerce server
  - Access control and authentication
  - Operating system controls
  - Firewalls.

Chapter subject summary

This chapter describes the security measures that can be used by companies to reduce or eliminate intellectual property theft. It describes how client computers can be protected from virus attacks and ill-intentioned programs. This chapter also details available protection mechanisms to secure information sent between a client computer and an electronic commerce server and how the integrity of messages sent between systems can be secured.

Protecting electronic commerce assets

A company will not be able to produce a secure electronic commerce system unless there is a written security policy. Such a security policy must detail:

- what assets are to be protected
- what is needed to protect these assets
- analysis of the likelihood of threats
- rules to be enforced to protect assets.

Guidelines state that a company should protect its assets from unauthorised: disclosure; modification; destruction. A typical rule that may be found in a security policy is 'do not reveal confidential information to anyone outside the company'.

What are the methods that can be used to collect payment from customers through electronic commerce applications? What are software wallets and smart cards? What is the SET protocol and can it help to ensure the security of credit card transactions?

Chapter 7

Electronic Payment Systems

Introduction

In this chapter you will learn the techniques and technologies used to implement electronic payment systems for electronic commerce applications. You will also discover the advantages and disadvantages of different payment systems such as electronic cash, electronic wallets, smart cards and credit card transactions. In addition, this chapter also outlines how the SET protocol is being developed to protect credit card transactions over the internet.

Learning objectives

In this chapter you will learn about:

- Four distinct methods to collect payments from customers
- How credit and debit card processing is handled for electronic commerce transactions
- How the SET protocol protects credit card transactions
- Software wallets and how they work
- The history and near-term future for electronic cash
- How electronic cash systems are implemented
- The role of smart cards in electronic commerce
- Which of the electronic payment systems currently is the most popular and which show promise of gaining acceptance.

Student reading

It is strongly advised that you now read Chapter 7 of the recommended course text.

'Electronic Commerce' James Perry & Gary Schneider, 2nd ed Paperback, 489 pages Course Technology, 2001 ISBN: 0619033789

Chapter outline

This chapter outline is provided as a guide to your reading for this chapter. If by chance you cannot obtain the recommended course text for this chapter, then you should aim to read texts which cover the following topics.

- The Basics of electronic payment systems
- Electronic cash
  - Holding electronic cash: online and off-line cash
  - The advantages and disadvantages of electronic cash
  - How electronic cash works

- Providing security for electronic cash
- Past and present systems
  1. Checkfree
  2. Clickshare
  3. Cybercash
  4. Digicash
  5. E-coin.net
  6. Millicent

- Electronic wallets
  - Agile wallet
  - E-wallet
  - Microsoft wallet
  - The W3C proposed standard
  - The ECML standard

- Smart Cards
  - What is a smart card?
  - Mondex smart card

- Credit and charge cards
  - Payment acceptance and processing
    1. Open and closed loop systems
    2. Setting up a merchant account
    3. Processing payment cards online
    4. Secure electronic transaction protocol

Chapter subject summary

This chapter outlines the key methods for collecting payments from customers online. It describes how electronic wallets operate and how other methods of payment are processed.

Electronic payment systems

The primary methods for electronic payments are:

- electronic cash
- electronic software wallets
- smart cards
- credit and debit cards
- digital cash minted by third party organisations.

Each of the above payment methods are now briefly described.

Digital Certificates & Encryption

The Need for Security

On the Internet, information you send from one computer to another passes through numerous systems before it reaches its destination. Normally, the users of these intermediary systems don't monitor the Internet traffic routed through them, but someone who's determined can intercept and eavesdrop on your private conversations or credit card exchanges. Worse still, they might replace your information with their own and send it back on its way.

Due to the architecture of the Internet and intranets, there will always be ways for unscrupulous people to intercept and replace data in transit. Without security precautions, users can be compromised when sending information over the Internet or an intranet. This has serious implications for Internet Commerce. For Internet Commerce to exist, there has to be a means to secure data sent over the Internet. Without a secure means of communication, commerce cannot exist.

How do I protect my data?

Encryption & Digital Certificates are the solution for Internet Commerce. Used together, they protect your data as it travels over the Internet.

Encryption is the process of using a mathematical algorithm to transform information into a format that can't be read (this format is called *cipher text*). Decryption is the process of using another algorithm to transform encrypted information back into a readable format (this format is called *plain text*).

Digital Certificates are your digital passport, an Internet ID. They are verification of you who you are and the integrity of your data.

Combined, encryption and digital certificates protect and secure your data in the following four ways:.

- ***Authentication:*** This is digital verification of who you are, much in the same way your driver's license proves your identity. It is very easy to send spoofed email. I can email anyone in the world pretending I am the President of the United States. Using standard email, there is no way to verify who the sender is, i.e. if it is actually the President. With digital signatures and certificates, you digitally encode verifiable proof of your identity into the email.
- ***Integrity:*** This is the verification that the data you sent has not been altered. When email or other data travels across the Internet, it routes through various gateways (way stations). It is possible for people to capture, alter, then resend the message. Example, your boss emails the company president stating that you should be fired. It is possible for you to intercept that email and change it saying you deserve a $10,000 raise. With digital certificates, your email cannot be altered without the recipient knowing.
- ***Encryption:*** This ensures that your data was unable to be read or utilized by any party while in transit. Your message is encrypted into incomprehensible gibberish before it leaves your computer. It maintains it encrypted (gibberish) state during it's travel through the Internet. It is not de-crypt until the recipient receives it. Because of the public-key cryptography used (discussed later) only the recipient can decipher the received message, no one else can.

- ***Token verification:*** Digital tokens replace your password which can be easily guessed. Tokens offer a more secure way of access to sensitive data. The most common way to secure data or a web site is with passwords. Before anyone accesses the data, they are prompted with their user login id and password. However, this is easily cracked using various security software (such as Crack 5.0, etc.). Also, passwords can be found with other means, such as social engineering. Passwords are not secure. Token verification is more secure. Your digital certificate is an encrypted file that sits on your hardrive. When you need access to a system, that system asks you for your digital certificate instead of a password. Your computer would then send the certificate, in encrypted format, through the Internet, authorizing you for access. For this to be compromised, someone would have to copy this file from your computer, AND know your password to de-crypt the file.

How does it all work?

Encryption

To understand how this all works, we need to start with the basics. Encryption has been around for centuries, Julius Caesar used encrypted notes to communicate with Rome thousands of years ago. This traditional cryptography is based on the sender and receiver of a message knowing and using the same secret key: the sender uses the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message. For Caesar, the letter A was represented by the letter D, B by the letter E, C by the letter F, etc. The recipient would know about this sequence, or key, and decrypt his message. This method is known as secret-key or symmetric cryptography. Its main problem is getting the sender and receiver to agree on the key without anyone else finding out. Both sides must find some "secure" way to agree or exchange this common key. Because all keys must remain secret, secret-key cryptography often has difficulty providing secure key management, especially in open systems with a large numbers of users, such as the Internet.

21 years ago, a revolution happened in cryptography that changed all this, public-key cryptography. In 1976, Whitfield Diffie and Martin Hellman, introduced this new method c encryption and key management. A public-key cryptosystem is a cryptographic system that uses a pair of unique keys (a public key and a private key). Each individual is assigned a pair of these keys to encrypt and decrypt information. A message encrypted by one of these keys can only be decrypted by the other key in the pair:

- The public key is available to others for use when encrypting information that will be sent to an individual. For example, people can use a person's public key to encrypt information they want to send to that person. Similarly, people can use the user's public key to decrypt information sent by that person.
- The private key is accessible only to the individual. The individual can use the private key to decrypt any messages encrypted with the public key. Similarly, the individual can use the private key to encrypt messages, so that the messages can only be decrypted with the corresponding public key.

What does this mean?

Exchanging keys is no longer a security concern. I have my public key and private key. I send my public key to anyone on the Internet. With that public key, they encrypt their email. Since the email was encrypted with my public key, ONLY I can decrypt that email with my private key, no

one else can. If I want to encrypt my email to anyone else on the Internet, I need their public key. Each individual involved needs their own public/private key combination.

Now, the big question is, when you initially receive someone's public key for the first time, how do you know it is them? If spoofing someone's identity is so easy, how do you knowingly exchange public keys, how do you TRUST the user is really who he says he is? You use your digital certificate. A digital certificate is a digital document that vouches for the identity and key ownership of an individual, a computer system (or a specific server running on that system), or an organization. For example, a user's certificate verifies that the user owns a particular public key. Certificates are issued by certificate authorities, or CAs. These authorities are responsible for verifying the identity and key ownership of the individual before issuing the certificate, such as ***Verisign***, http://www.verisign.com.

Authentication & Integrity

We now have a secure means of encrypting data, one of the four methods of securing data on the Internet. Two others, authentication and data integrity, are combined in what is called a digital signature. A digital signature works as follows:

- ***Authentication:*** a specific individual sent a message (in other words, no impersonator claiming to be the individual sent the message).
- ***Integrity:*** this particular message was sent by the individual (in other words, no one altered the message before it was received).

When you email someone, your public/private key combination creates the digital signature. It does this using the following format:

1. The sender uses a *message-digest algorithm* to generate a shorter version of the message that can be encrypted. This shorter version is called a *message digest.* Message digests and message-digest algorithms are explained in the next section.
2. The sender uses their private key to encrypt the message digest.
3. The sender transmits the message and the encrypted message digest to the recipient.
4. Upon receiving the message, the recipient decrypts the message digest.
5. The recipient uses the hash function on the message to generate the message digest.
6. The recipient compares the decrypted message digest against the newly generated message digest.
    - If the message digests are identical, the recipient knows that the message was indeed sent by the person claiming to be the sender and that the message was not modified during transmission.
    - If the message digests differ, the recipient knows that either the message was sent by someone else claiming to be the sender or that the message was modified or damaged during transmission.

The encrypted message digest serves as a digital signature for the message. The signature verifies the identity of the sender and the contents of the message.

If the message is sent by someone claiming to be the sender, this person does not have access to the sender's private key. The person claiming to be the sender must use a different private key to encrypt the message digest.

Because the recipient uses the sender's public key to decrypt the message digest (and not the actual public key corresponding to the private key used to encrypt the message digest), the decrypted message digest will not match the newly generated message digest.

If the message was modified during transmission, the hash function will generate a different message digest when applied after the transmission.

Tokens

Tokens represent the fourth security option by replacing passwords. Tokens are simply your digital certificate residing on your hardrive. When a computer prompts you for your password, your computer sends your certificate over the Internet instead. Your certificate verifies your identity instead of the password. This is a more secure (and easier) means of verificatio

How Secure is all this?

Just how secure is encryption. The strength of encryption is measured in bits, or how big the key is. The bigger the key, the stronger the encryption. There are currently 3 commonly used key sizes used commercially, 40, 56, and 128 bit. Originally, the government allowed only 40 bit keys for exportation. However, this proved far to weak for security. In February of 1997, a college student was able to crack 40 bit encrypted data within 4 hours (http://www2.ecst.csuchico.edu/~atman).

> Berkeley -- It took UC Berkeley graduate student Ian Goldberg only three and a half hours to crack the most secure level of encryption that the federal government allows U.S. companies to export.
>
> Yesterday (1/28) RSA Data Security Inc. challenged the world to decipher a message encrypted with its RC5 symmetric stream cipher, using a 40-bit key, the longest keysize allowed for export. RSA offered a $1,000 reward, designed to stimulate research and practical experience with the security of today's codes.
>
> Goldberg succeeded a mere 3 1/2 hours after the contest began, which provides very strong evidence that 40-bit ciphers are totally unsuitable for practical security.

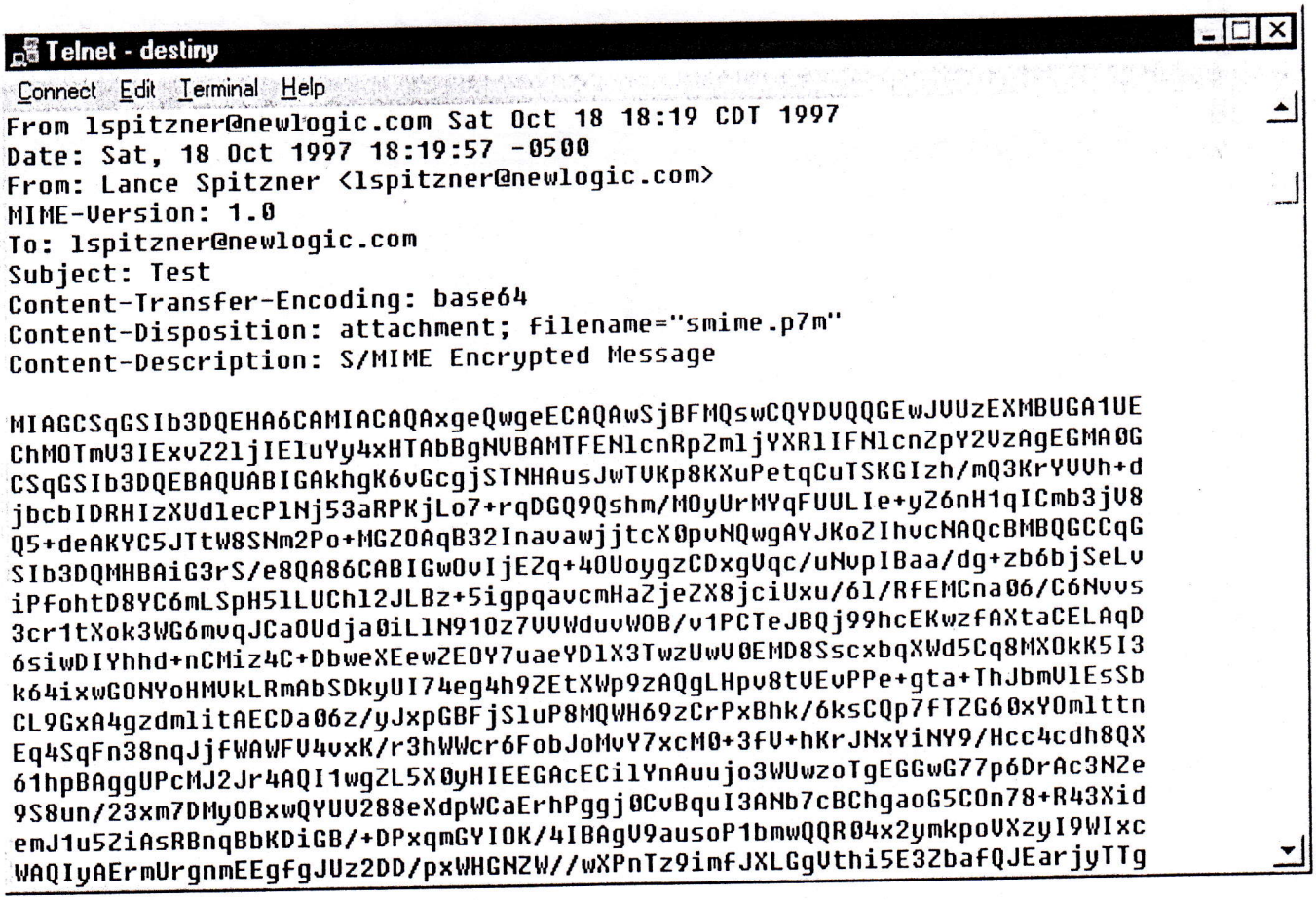
In June of 1997, a organized group of people were able to crack 56 bit DES encryption in 140 days. This group shared their resources throughout the Internet utilizing software called DESCHALL (http://www.rsa.com/des).

> With a possible 72 quadrillion keys to test, this distributed attack would require an incredibly large amount of computing power. And compute the DESCHALL team did, at some points testing almost seven billion keys per second.
>
> In the end, the DESCHALL effort solved the DES challenge after only searching 24.6% of the key space. (about 18 quadrillion keys!) The winning key was determined by Michael Sanders, using a Pentium 90 MHz desktop PC with 16 megs of RAM.

Many believe this security is good enough. By the time your data can be compromised, (3 months) it is of little value because it took so long. However, to truly ensure the security of your data, most Internet Commerce uses 128 bit encryption. Keep in mind, key strength increases exponentially, making 128 bit encryption thousands of times more difficult to compromise. Because of its strength, the government has prohibited its exportation, it can only be used within the United States. At this time, no one has cracked this encryption. 128 bit encryption is expected to remain secure well past the year 2000.

What it Looks Like

Below is an example of a message that has been encrypted and signed, but intercepted before the recipient has received it. Notice how the body of the entire message is "gibberish", i.e., the message cannot be read. That is what encryption looks like.



Below is an example of the same message, but received by the intended recipient. The recipient has decrypted the message and verified the message's integrity & /authenticity. The protocol or Internet standard used for Digital Certificates is X.509 & S/MIME. Any email system that has these open based standards can use Digital Certifcates for Internet Commerce. The image below is of Netscape Navigator, which is both X.509 and S/MIME compliant.

Public and Private Key Encryption Systems

Private and public keys are used in two main encryption systems: Symmetric and Asymmetric.

Symmetric Encryption

This system uses only private keys. This requires the private key (code) to be installed on specific computers that will be used for exchanging messages between certain users. The system works pretty much like two best friends using a decoder ring to send secret messages to each other. Both friends know which code they are using and thus, only they will have the key to crack and encode secret messages.

Asymmetric Encryption

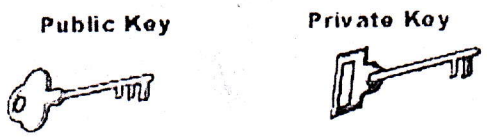
The Asymmetric Encryption system uses both the private and public keys. The private key is for you and the public key is published on line for others to see.

They use the public key to access the encryption code that corresponds to your private key. So, if you are sending an encrypted message to Susan which you do not want others to see, you would use her public key to encrypt it. She will be able to decrypt it with her own corresponding private key. Likewise, if she sends a message to you, she uses your public key to encrypt the message and you would use your private key to decrypt it.

The Public and Private key pair comprise of two uniquely related cryptographic keys (basically long random numbers). Below is an example of a Public Key:

3048 0241 00C9 18FA CF8D EB2D EFD5 FD37 89B9 E069 EA97 FC20 5E35 F577 EE31 C4FB C6E4 4811 7D86 BC8F BAFA 362F 922B F01B 2F40 C744 2654 C0DD 2881 D673 CA2B 4003 C266 E2CD CB02 0301 0001.

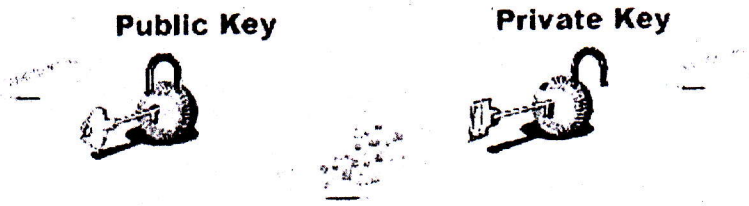
The Public Key is what its name suggests - Public. It is made available to everyone via a publicly accessible repository or directory. On the other hand, the Private Key must remain confidential to its respective owner.



Because the key pair is mathematically related, whatever is encrypted with a Public Key may only be decrypted by its corresponding Private Key and vice versa.

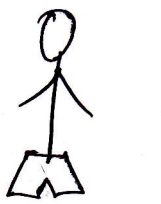
For example, if Bob wants to send sensitive data to Alice, and wants to be sure that only Alice may be able to read it, he will encrypt the data with Alice's Public Key. Only Alice has access to

her corresponding Private Key and as a result is the only person with the capability of decrypting the encrypted data back into its original form.



As only Alice has access to her Private Key, it is possible that only Alice can decrypt the encrypted data. Even if someone else gains access to the encrypted data, it will remain confidential as they should not have access to Alice's Private Key.

Public Key Cryptography can therefore achieve Confidentiality. However another important aspect of Public Key Cryptography is its ability to create a Digital Id or Signature. Digital IDs are ideal because they can't be easily forged. They allow for a more electronically secured communication line because they enable you to make use of these encryption systems. By applying them to your PDF files, you are ensuring their security.



Bob

Alice

3. Encryption

Encryption is the process of changing text so that it is no longer easy to read. A very simple example is the following sentence:

```
Guvf vf n fvzcyr fhofgvghgvba pvcure.
```

Commercial encryption uses methods which are a lot more secure than the one I used to produce that example. Almost all modern encryption methods rely on a key - a particular number or string of characters which are used to encrypt, decrypt, or both.

In the next sections, common encryption methods are presented. To illustrate how they work, ficticious characters named Bob and Alice will be introduced. Private key encryption and public key encryption are discussed, as are their limitations.

3.1. Private key encryption

Private key encryption is the standard form. Both parties share an encryption key, and the encryption key is also the one used to decrypt the message. The difficulty is sharing the key before you start encrypting the message - how do you safely transmit it?

Many private key encryption methods use public key encryption to transmit the private key for each data transfer session.

If Bob and Alice want to use private key encryption to share a secret message, they would each use a copy of the same key. Bob writes his message to Alice and uses their shared private key to encrypt the message. The message is then sent to Alice. Alice uses her copy of the private key to decrypt the message. Private key encryption is like making copies of a key. Anyone with a copy can open the lock. In the case of Bob and Alice, their keys would be guarded closely because they can both encrypt and decrypt messages.

3.2. Public Key encryption

Public key encryption uses two keys - one to encrypt, and one to decrypt. The sender asks the receiver for the encryption key, encrypts the message, and sends the encrypted message to the receiver. Only the receiver can then decrypt the message - even the sender cannot read the encrypted message.

When Bob wants to share a secret with Alice using public key encryption, he first asks Alice for her public key. Next, Bob uses Alice's public key to encrypt the

message. In public key encryption, only Alice's private key can unlock the message encrypted with her public key. Bob sends his message to Alice. Alice uses her private key to decrypt Bob's message.

The things that make public key encryption work is that Alice very closely guards her private key and freely distributes her public key. She knows that it will unlock any message encrypted with her public key.

3.3. Limitations of encryption

Cryptanalysis, or the process of attempting to read the encrypted message without the key, is very much easier with modern computers than it has ever been before. Modern computers are fast enough to allow for 'brute force' methods of cryptanalysis - or using every possible key in turn until the 'plain text' version of the message is found.

The longer the key, the longer it takes to use the 'brute force' method of cryptanalysis - but it also makes the process of encrypting and decrypting the message slower. Key length is very important to the security of the encryption method - but the 'safe' key length changes every time CPU manufacturers bring out a new processor.

Encryption does not make your data secure. Not using encryption, however, means that any data in transit is as easy to read as the contents of a postcard, sent in regular mail. Encryption at least ensures that anyone who does read your messages has worked hard at it.

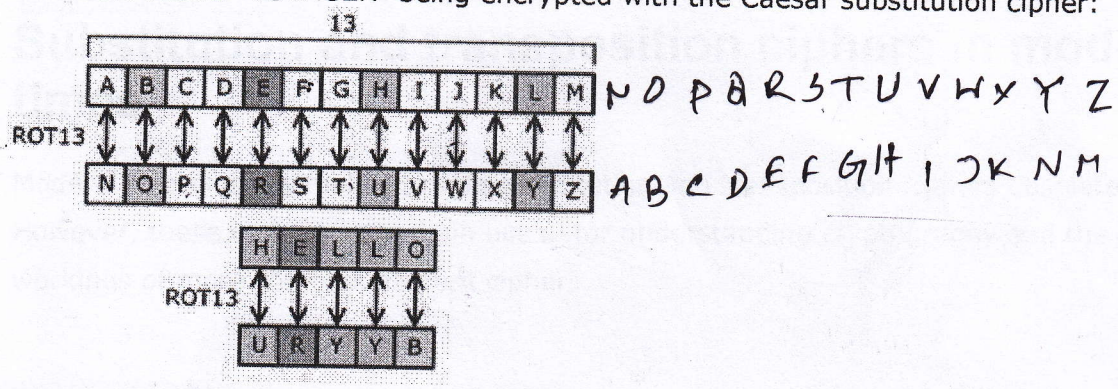
Substitution and Transposition Ciphers

Substitution and transposition ciphers are two categories of ciphers used in classical cryptography. Substitution and transposition differ in how chunks of the message are handled by the encryption process.

Substitution ciphers

Substitution ciphers encrypt plaintext by changing the plaintext one piece at a time. The Caesar Cipher was an early substitution cipher. In the Caesar Cipher, each character is shifted three places up. Therefore, A becomes D and B becomes E, etc...

This table shows "VOYAGER" being encrypted with the Caesar substitution cipher:



Plaintext	V	O	Y	A	G	E	R
Key	+3	+3	+3	+3	+3	+3	+3
Ciphertext	Y	R	B	D	J	H	U

A more complex substitution cipher would be created if, instead of incrementing each character by three, we used a more complex key. This table shows a simple substitution cipher with a key of "123".

Plaintext	V	O	Y	A	G	E	R
Key	+1	+2	+3	+1	+2	+3	+1
Ciphertext	W	Q	B	B	I	H	S

An even more complex substitution cipher can be made by having each character of the alphabet correspond to a different letter of the alphabet, without a set pattern.

Plaintext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Key	T	O	E	U	N	Z	I	A	G	X	P	Q	Y	R	H	V	S	M	D	F	C	J	W	B	K	L

Using this substitution cipher to encrypt VOYAGER would give us these results:

Plaintext	V	O	Y	A	G	E	R
Ciphertext	J	H	K	T	X	N	M

The Vernam Cipher, or one time pad, is a simple substitution cipher where the key length equals the message length.

ROT-1 is a simple substitution cipher used to encode messages on Usenet.

Transposition ciphers

Transposition ciphers encrypt plaintext by moving small pieces of the message around. Anagrams are a primitive transposition cipher.

This table shows "VOYAGER" being encrypted with a primitive transposition cipher where every two letters are switched with each other:

V	O	Y	A	G	E	R
O	V	A	Y	E	G	R

Substitution and transposition ciphers in modern times

Modern cryptanalysis makes simple substitution and transposition ciphers obsolete. However, these techniques remain useful for understanding cryptography and the workings of more complex modern ciphers.

1.5 Identity and keys

Until now, we have taken for granted the keys being used for encryption/decryption and digital signature/verification belong to Bob and Alice. How can we be sure that Alice is really Alice? And, how can Alice be sure that only Bob will see what she encrypted? So far, the only thing we know is that the user of a given key pair has signed and encrypted the message. But, is he really the owner? George, for instance, may have sent a message to Bob pretending that he is Alice; Bob cannot tell whether or not it is Alice or George who is sending the message. The same applies to Bob's public-key. This issue is solved by the use of certificates.

2. What is a Certificate

A certificate is a piece of information that proves the identity of a public-key's owner. Like a passport, a certificate provides recognized proof of a person's (or entity) identity. Certificates are signed and delivered securely by a trusted third party entity called a Certificate Authority (CA). As long as Bob and Alice trust this third party, the CA, they can be assured that the keys belong to the persons they claim to be.

A certificate contains among other things:

1) The CA's identity
2) The owner's identity
3) The owner's public-key
4) The certificate expiry date
5) The CA's signature of that certificate
6) Other information that is beyond the scope of this article.

With a certificate instead of a public-key, a recipient can now verify a few things about the issuer to make sure that the certificate is valid and belongs to the person claiming its ownership:

1) Compare the owner's identity
2) Verify that the certificate is still valid
3) Verify that the certificate has been signed by a trusted CA
4) Verify the issuer's certificate signature, hence making sure it has not been altered.

Bob can now verify Alice's certificate and be assured that it is Alice's private-key that has been used to sign the message. Alice must be careful with her private-key and must not divulge how to get to it; by doing so, she is enforcing one aspect of the non-repudiation feature associated with her digital signature. As will be seen in section 3.2, there is more to consider for effective non-repudiation support.

Note that certificates are signed by a CA, which means that they cannot be altered. In turn, the CA signature can be verified using that CA's certificate.

2.1 Certificate validation added to the process

When Alice encrypts a message for Bob, she uses Bob's certificate. Prior to using the public-key included in Bob's certificate, some additional steps are performed to validate Bob's certificate:

1) Validity period of Bob's certificate
2) The certificate belongs to Bob
3) Bob's certificate has not been altered
4) Bob's certificate has been signed by a trusted CA

Additional steps would be required to validate the CA's certificate in the case where Alice does not trust Bob's CA. These steps are identical to the ones requires to validate Bob's certificate. In the example below, it is assumed that both Bob and Alice trust that CA.

2.2 Beyond the mechanics

So far, this article covered in some details the public-key mechanics associated with encryption and digital signature. In section 2.1 the notion of Certificate Authority has been brought up. The CA is the heart of a Public-Key Infrastructure (PKI).

3. What is a PKI

A PKI is a combination of software and procedures providing a means for managing keys and certificates, and using them efficiently. Just recall the complexity of the operations described earlier in this article for having a feel on the absolute necessity to provide users with appropriate software support for encryption and digital signature. But nothing has been said yet about management.

3.1 Key and certificate management

Key and certificate management is the set of operations required to create and maintain keys and certificates. The following is the list of the major points being addressed in a managed PKI:

1) **Key and certificate creation:** How to generate key pairs? How to issue certificates to the users? A PKI must offer software support for key pair generation as well as certificate requests. In addition, procedures must be put in place to verify the user identity prior to allowing him to request a certificate.

2) **Private-key protection:** How will the user protect his private-key against misuse by other malicious users?
Certificates are widely accessible because they are used for either encryption or signature verification. Private-keys require some reasonable level of protection because they are used either for decryption or for digital signature. A strong password mechanism must be part of the features of an effective PKI.

3) **Certificate revocation:** How to handle the situation where a user's private-key has been compromised? Similarly, how to handle the situation where an employee leaves the company? How to know whether or not a certificate has been revoked?
A PKI must provide a means by which a certificate can be revoked. Once revoked, this certificate must be included in a revocation list that is available to all users. A mechanism must be provided to verify that revocation list and refuse to use a revoked certificate.

4) **Key backup and recovery:** What happens to encrypted files when a user loses his private-key? Without key backup, all messages and files that have been encrypted with his public-key can no longer be decrypted and are lost forever. A PKI must offer private-key backup and a private-key recovery mechanism such that the user can get back his private-key to be able to get access to his files[11].

5) **Key and certificate update:** What happens when a certificate reaches or is near its expiry date? Keys and certificates have a finite lifetime. A PKI must offer a mechanism to at least update the expiry date for that certificate. Good practice though is to update the user's keys and certificates. The key and certificate update can be automatic in which case the end user gets notified that his keys have been updated, or can require that the user performs an action during or before his keys and certificates expire; if this case, the PKI must inform the user that this action is required prior the expiry time of his keys and certificates.

6) **Key history management:** After several key updates, how will a user decide which private-key to use to decrypt files?
Each key update operation generates new key pairs. Files that have been encrypted with previous public-keys can only be decrypted with their associated private-keys. Without key history management, the user would have to make decision on the key to use for decrypting files[12].