login:

password:

login

forums

free classifieds

logged users ::
active for last 5 minutes

Contributors
mikecc

Standard users
chelmon
drgnfly
hanibaal
killerdark
lack
pikardo
prOBEX
rompep
Sharth
viko21

registered users:106131

select a language

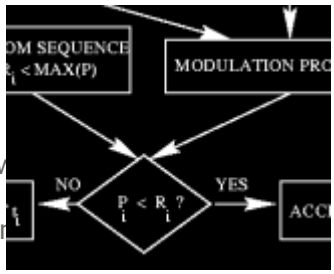
English / English

submit

Understanding the basics of algorithms and basic security applications.

@ Articles -> Encryption May 28 2002 - 20:32 EST

gregory_a_e writes: Simply put, this article is a brief overview of algorithms. What they are, how they tie into computers.



By no means is this a technical article, nor an obscure tutorial of some sort, so if you're expecting to learn the specifics of cryptography and algorithms, I'm sorry to say you won't find it here. If you do become interested in the subject, I would suggest some reading material that is suitable. Applied Cryptography and Digital Security in a Networked world by Bruce Schneier, Introduction to algorithms which was compiled by a vast board of academics (a broad and extremely dry and technical book, not the first one you should read).

Now that I've made that clear, let's get on with the article.

What are algorithms?

An algorithm is any computational procedure, or protocol, that begins with certain set value (x) and produces some

features

post news

New Order **forums**

online chat
(irc.box.sk / #neworder)

For more discussion boards check disc.box.sk

file and links archive

free classifieds

select a language

English / English

submit

articles

Articles

submit

themes of the month

- **Does capital drive the Internet?**
  Apr 02 2002 - 11:46
- **Securing Your Windows PC**
  Feb 11 2002 - 18:05
- **Cryptography - RSA and Quants**
  Jan 10 2002 - 16:11
- **Ethical Hacking**
  Nov 18 2001 - 13:03
- **Programming Languages and their History**
  Nov 09 2001 - 14:20
- **Denial of service**
  Oct 17 2001 - 10:15
- **Computer Viruses**
  Sep 14 2001 - 19:55

quotable quotes

How do you spell Yahoo?
saruman (tech. support call)

other set value (y). The algorithm is the process by which x becomes y.

Algorithms have quite a few uses in computers:

A. Algorithms are used to manipulate and manage the massive heap of data transferred through this automation, this wonderful system, we call the internet.

B. Fixing computational problems between x and y relations by analyzing the set specifications vs. the specifications where any problems might occur.

C. Cryptography and digital signatures depend on number theory, and determined, specific algorithms. Meaning that half of the security of the net, at least as far as information trading, and transactions go, depends on algorithms, mathematics in effect, to provide the proper level of security. This is a sensitive thing for several reasons. One, algorithms are highly difficult to successfully produce, and too many of them are filled with holes, and are fairly unstable. And it should be noted that one of the biggest risks to security is unstable algorithms in sensitive settings/mediums.

D. Algorithms are used in building programs to determine efficiency, distribution of work, and the like in industrial and manufacturing settings. Basically algorithms are written to illustrate ideal work and the work in practice vs. the work in the ideal algorithms are compared and this is often how efficiency can be decided, improved, and mathematically represented.

There are other purposes to

- **Anonymity on the web**
  Aug 2001

newsletter

#7 - Dec 01/Jan 02
#6 - Nov, 10 2001
#5 - Oct, 12 2001
#4 - Sep, 3 2001
#3 - Aug, 10 2001
#2 - Jul, 21 2001
#1 - Jul, 10 2001

New Order FAQ

Edge engine -> code
submit

linking & backends

Information about how to link to NewOrder.

New Order news backend or more advanced version.

glossaries

search

merchandise

Get your very own New Order, Unhackable Tanktop, Cult of the Dead Rabbit, or Astalavista shirts now!

algorithms, but not ones that are quite as common, or integral as the ones listed above.

The fact is algorithms are everywhere, even the human mind, is based on some obscure, inefficient, and undiscovered algorithm. It's a fact that security widely depends on it, and it's a fact that it has practical uses in the modern world. This is the most important part, what makes having a basic understanding of it, so important.

Standards of algorithms -
1. The algorithm has to be based on mathematical functions that are difficult to reverse engineer (integer factorization, reversing exponential values, any integers written in scientific notation can be easily represented, but not easily discovered, etc.).

2. The human component has to keep any keys, codes, and information that is meant to be secret, secret. (This is no doubt the biggest problem, and why social engineering is so dangerous. Everyone knows this, and smart people know to prevent it, and smart hackers, they take advantage of it).

3. The algorithms have to be constructed in such a way that brute force attacks are only effective vs. plaintext keys and visible cyphertext (this implies that both the plaintext and cyphertext have to be hidden so as to not be revealed.

4. That a plaintext and cyphertext copy cannot be both readily available to any parties that might want to intercept them. (They can easily be compared and eventually decoded. Same as what I mentioned in step 3).

Probably the most widespread use for cryptography is in internet security. It's not the cure for all the problems, but it's important, it is essential to the survival of business and e-commerce most importantly.

A. Symmetric Encryption -
The main idea behind cryptography, is that the message, the plaintext, is sent securely, and is turned into cyphertext gibberish and incomprehensible symbols so that only the one that it is meant for can read it (after it is deciphered).

For this to work the cryptography has to be good from two angles. One, the algorithm has to be good, so that it cannot be reverse-engineered, or cracked. Two, that it cannot be universal, and thus can only work within a certain medium (I.E. a certain language, or only with certain software, etc.) This makes creating good algorithms difficult, and stressful, and making them universal, impossible.

OK, simply put the idea behind this type of encryption is simple, person A can switch data with person B, and big brother C can't read it. This is where the public-key encryption comes in. The algorithm and process is known to everybody, but the key, ah the wonderful key, therein lies the trick. Assuming the algorithm is secure, then the only way to turn the cyphertext back into plaintext is by unlocking it with the key. The key is essentially a (preferably) secret plaintext password that decrypts the message.

This kind of thing is simple, and assuming people can be trusted,

- Multiple Red-M 1050 Blue Tooth Access Point Vulnerabilities
- SHOUTcast Remote Buffer Overflow (icy-name)

Jun, 03 2002

- Mnews Exploit Code
- Remote Quake Server CVAR Leak
- Multiple Vulnerabilities in Novell Netware
- Courier CPU Exhaustion (Negative Year)
- Internet Explorer DoS (window.open)

Jun, 02 2002

- Quantum SNAP Server DoS and Sequence Number Vulnerability
- Security Vulnerability in ECS-K7S5A(L) Boards
- AIM+ Found to Contain a SpyWare
- Multiple Security Vulnerabilities in QNX (dumper, monitor, crttrap)
- Informix SE /lib/sqlexec Security Vulnerability (INFORMIXDIR)
- CGIscript.net - csPassword.cgi - Multiple Vulnerabilities
- Shambala Server Directory Traversal and DoS

view all exploits

pretty secure.

It of course gets more complicate the more people are involved, but the idea generally remains the same.

B. MACs or message authentication codes -
MACs are the checkers of cryptography. They don't have anything to do with cryptography, but have their own algorithms.

The idea behind MACs is extremely simple, when information, packets, are sent, they are tagged with a code, and the MAC recognizes the code so that it can authenticate the source and confirm that the packet comes from the place where it's supposed to come from, and that it should, theoretically, contain the proper data.

It is my opinion that this process is painfully obvious, and painfully obviously necessary. And, if you think about it, a fairly ingenious, and almost completely error-proof verification method, assuming that the human component is not in error, because the technology is almost impossible to be.

C. Hash-Functions -
The idea behind Hash functions is that they are mathematically computer bits of data that are easily read to confirm something, but impossible to reverse engineer so as to forge the source.

This could be used in various ways. Say you came upon some source code, there is nothing to lead you to who made it. Now say you receive a second copy,

who's to say which is the real program, you could compile and run it, but then you risk possible damage. This is where the has function comes in. If you have the hash function for the real book, see which one it matches with, it's that simple. (Of course you'd have to have a copy of the Hash, distribution of hashes is not widespread yet, but it's certainly something to think about. And of course, I realize very few people would be stupid enough to compile programs w/o analyzing the code, etc.).

Well, as many people may have noticed, this was as I stated a very basic article, I hope those of you new to the field and the concepts learned something here, and that those of you who are experts in the field, well, I just hope I didn't make any noticeable, technical mistakes, though I'm quite sure I did.

read comments (20) | write comment| views: 3967 printer-friendly version