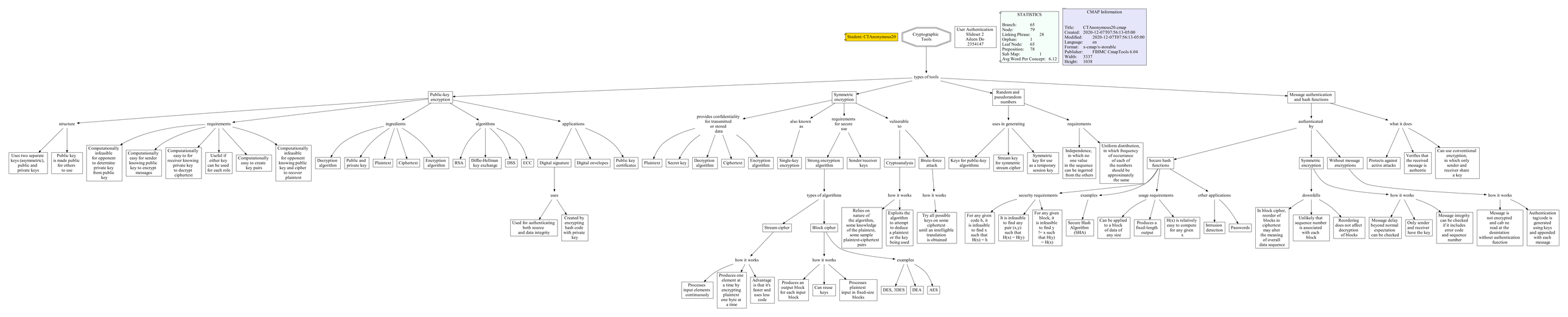
Student: CTAnonymous20
Cryptographic Tools
User Authentication
Slideset 2
Aileen Do
2354147
STATISTICS
Branch: 65
Node: 79
Linking Phrase: 28
Orphan: 1
Leaf Node: 65
Preposition: 78
Sub Map: 1
Avg Word Per Concept: 6.12
CMAP Information
Title: CTAnonymous20.cmap
Created: 2020-12-07T07:56:13-05:00
Modified: 2020-12-07T07:56:13-05:00
Language: en
Format: x-cmap/x-storable
Publisher: FIHMC CmapTools 6.04
Width: 3337
Height: 1038
types of tools
Public-key encryption
structure
Uses two separate keys (asymmetric), public and private keys
Public key is made public for others to use
requirements
Computationally infeasible for opponent to determine private key from public key
Computationally easy for sender knowing public key to encrypt messages
Computationally easy to for receiver knowing private key to decrypt ciphertext
Useful if either key can be used for each role
Computationally easy to create key pairs
Computationally infeasible for opponent knowing public key and cipher to recover plaintext
ingredients
Decryption algorithm
Public and private key
Plaintext
Ciphertext
Encryption algorithm
algorithms
RSA
Diffie-Hellman key exchange
DSS
ECC
applications
Digital signature
uses
Used for authenticating both source and data integrity
Created by encrypting hash code with private key
Digital envelopes
Public key certificates
Symmetric encryption
provides confidentiality for transmitted or stored data
Plaintext
Secret key
Decryption algorithm
Ciphertext
Encryption algorithm
also known as
Single-key encryption
requirements for secure use
Strong encryption algorithm
types of algorithms
Stream cipher
how it works
Processes input elements continuously
Produces one element at a time by encrypting plaintext one byte at a time
Advantage is that it's faster and uses less code
Block cipher
how it works
Produces an output block for each input block
Can reuse keys
Processes plaintext input in fixed-size blocks
examples
DES, 3DES
DEA
AES
Sender/receiver keys
vulnerable to
Cryptoanalysis
how it works
Relies on nature of the algorithm, some knowledge of the plaintext, some sample plaintext-ciphertext pairs
Exploits the algorithm to attempt to deduce a plaintext or the key being used
Brute-force attack
how it works
Try all possible keys on some ciphertext until an intelligible translation is obtained
Random and pseudorandom numbers
uses in generating
Keys for public-key algorithms
Stream key for symmetric stream cipher
Symmetric key for use as a temporary session key
requirements
Independence, in which no one value in the sequence can be ingerred from the others
Uniform distribution, in which frequency of occurrance of each of the numbers should be approximately the same
security requirements
For any given code h, it is infeasible to find x such that H(x) = h
It is infeasible to find any pair (x,y) such that H(x) = H(y)
For any given block, it is infeasible to find y != x such that H(y) = H(x)
examples
Secure Hash Algorithm (SHA)
usage requirements
Can be applied to a block of data of any size
Produces a fixed-length output
H(x) is relatively easy to compute for any given x
other applications
Intrusion detection
Passwords
Message authentication and hash functions
authenticated by
Secure hash functions
Symmetric encryption
downfalls
In block cipher, reorder of blocks in ciphertext may alter the meaning of overall data sequence
Unlikely that sequence number is associated with each block
Reordering does not affect decryption of blocks
how it works
Message delay beyond normal expectation can be checked
Only sender and receiver have the key
Message integrity can be checked if it includes error code and sequence number
Without message encryptions
how it works
Message is not encrypted and cab ne read at the desintation without authentication function
Authentication tag/code is generated using keys and appended with each message
what it does
Protects against active attacks
Verifies that the received message is authentic
Can use conventional encryption, in which only sender and receiver share a key