More on Cryptography
CS 136
Computer Security
Peter Reiher
February 5, 2008

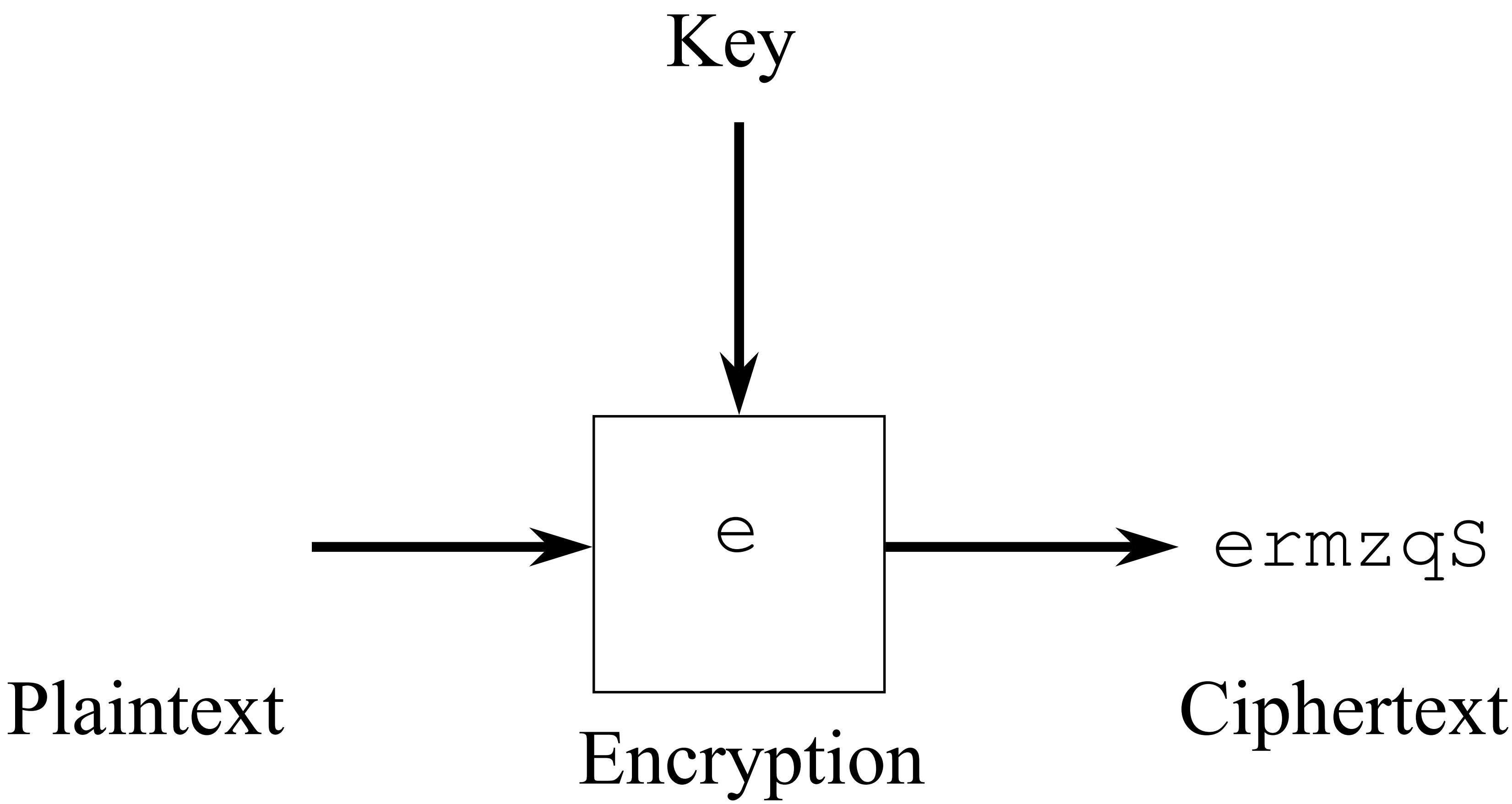
Outline

- Stream and block ciphers
- Desirable characteristics of ciphers
- Uses of cryptography
- Symmetric and asymmetric cryptography
- Digital signatures
- Secure hashes

Stream and Block Ciphers

- Stream ciphers convert one symbol of plaintext immediately into one symbol of ciphertext
- Block ciphers work on a given sized chunk of data at a time

Stream Ciphers



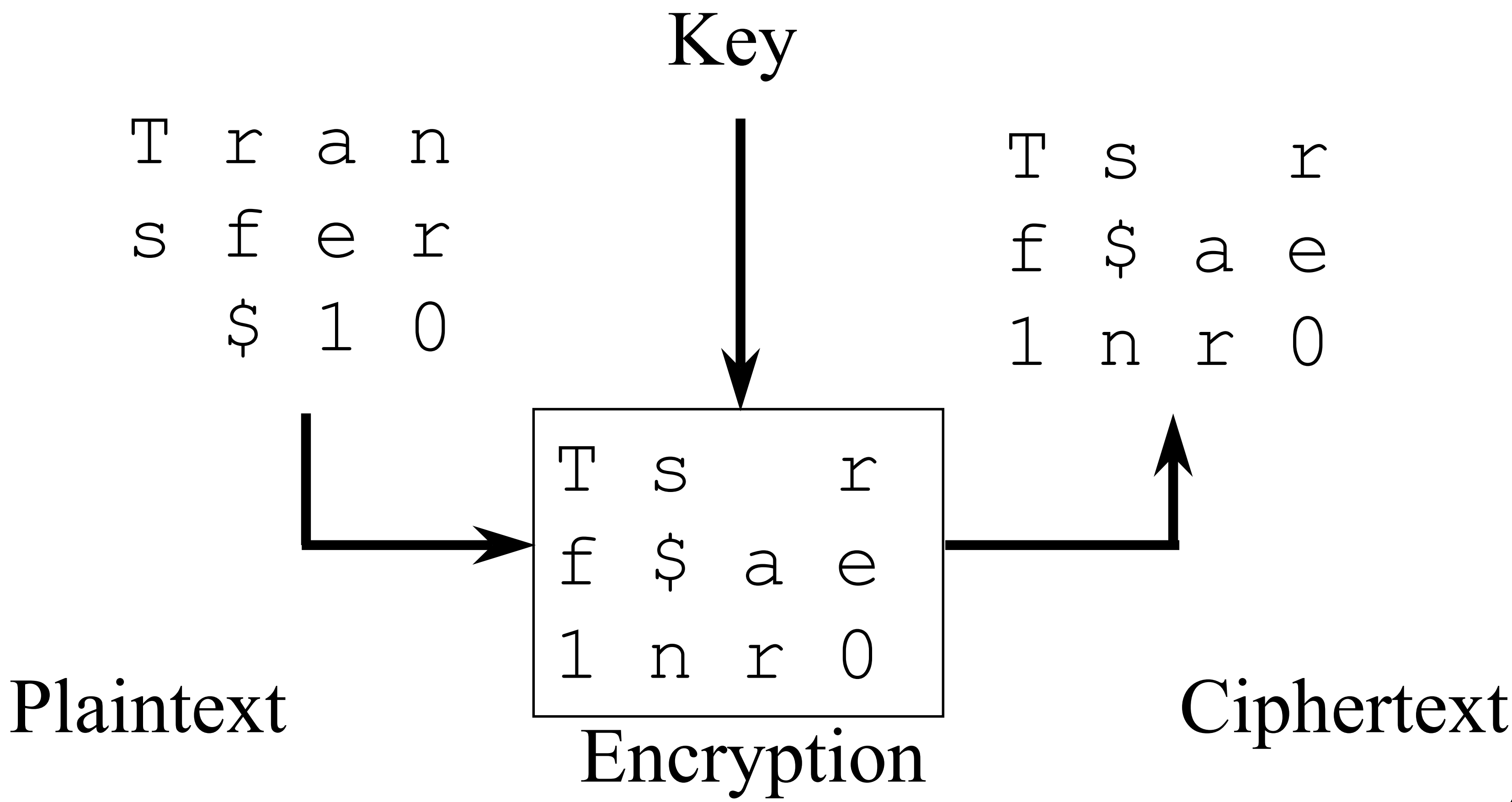
Advantages of Stream Ciphers

+ Speed of encryption and decryption
  - Each symbol encrypted as soon as it's available
+ Low error propagation
  - Errors affect only the symbol where the error occurred

Disadvantages of Stream Ciphers

- Low diffusion
  - Each symbol separately encrypted
  - Each ciphertext symbol only contains information about one plaintext symbol
- Susceptible to insertions and modifications
- Not good match for many common uses of cryptography

Block Ciphers



Advantages of Block Ciphers

+ Diffusion
  - Easier to make a set of encrypted characters depend on each other
+ Immunity to insertions
  - Encrypted text arrives in known lengths

Most common Internet crypto done with block cyphers

Disadvantages of Block Ciphers

- Slower
  - Need to wait for block of data before encryption/decryption starts
- Worse error propagation
  - Errors affect entire blocks

Desirable Characteristics of Ciphers

- Well matched to requirements of application
  - Amount of secrecy required should match labor to achieve it
- Freedom from complexity
  - The more complex algorithms or key choices are, the worse

More Characteristics

- Simplicity of implementation
  - Seemingly more important for hand ciphering
  - But relates to probability of errors in computer implementations
- Errors should not propagate

Yet More Characteristics

- Ciphertext size should be same as plaintext size
- Encryption should maximize *confusion*
  - Relation between plaintext and ciphertext should be complex
- Encryption should maximize *diffusion*
  - Plaintext information should be distributed throughout ciphertext

Uses of Cryptography

- What can we use cryptography for?
- Lots of things
  - Secrecy
  - Authentication
  - Prevention of alteration

Cryptography and Secrecy

- Pretty obvious
- Only those knowing the proper keys can decrypt the message
  - Thus preserving secrecy
- Used cleverly, it can provide other forms of secrecy

Cryptography and Authentication

- How can I prove to you that I created a piece of data?
- What if I give you the data in encrypted form?
  – Using a key only you and I know
- Then only you or I could have created it
  – Unless one of us told someone else the key . . .

Some Limitations on Cryptography and Authentication

- If both parties cooperative, cryptography can authenticate
  - Problems with non-repudiation, though
- What if three parties want to share a key?
  - No longer certain who created anything
  - Public key cryptography can solve this problem
- What if I want to prove authenticity <u>without</u> secrecy?

Cryptography and Non-Alterability

- Changing one bit of an encrypted message completely garbles it
  - For many forms of cryptography
- If a checksum is part of encrypted data, that's detectable
- If you don't need secrecy, can get the same effect
  - By encrypting only the checksum

Cryptography and Zero-Knowledge Proofs

- With really clever use, cryptography can be used to prove I know a secret
  - Without telling you the secret
- Seems like magic, but it can work
- Basically, using multiple iterations of cryptography in very clever ways

Symmetric and Asymmetric Cryptosystems

- Symmetric - the encrypter and decrypter share a secret key
  - Used for both encrypting and decrypting
- Asymmetric – encrypter has different key than decrypter

Description of Symmetric Systems

- $C = E(K,P)$
- $P = D(K,C)$
- *E()* and *D()* are not necessarily the same operations

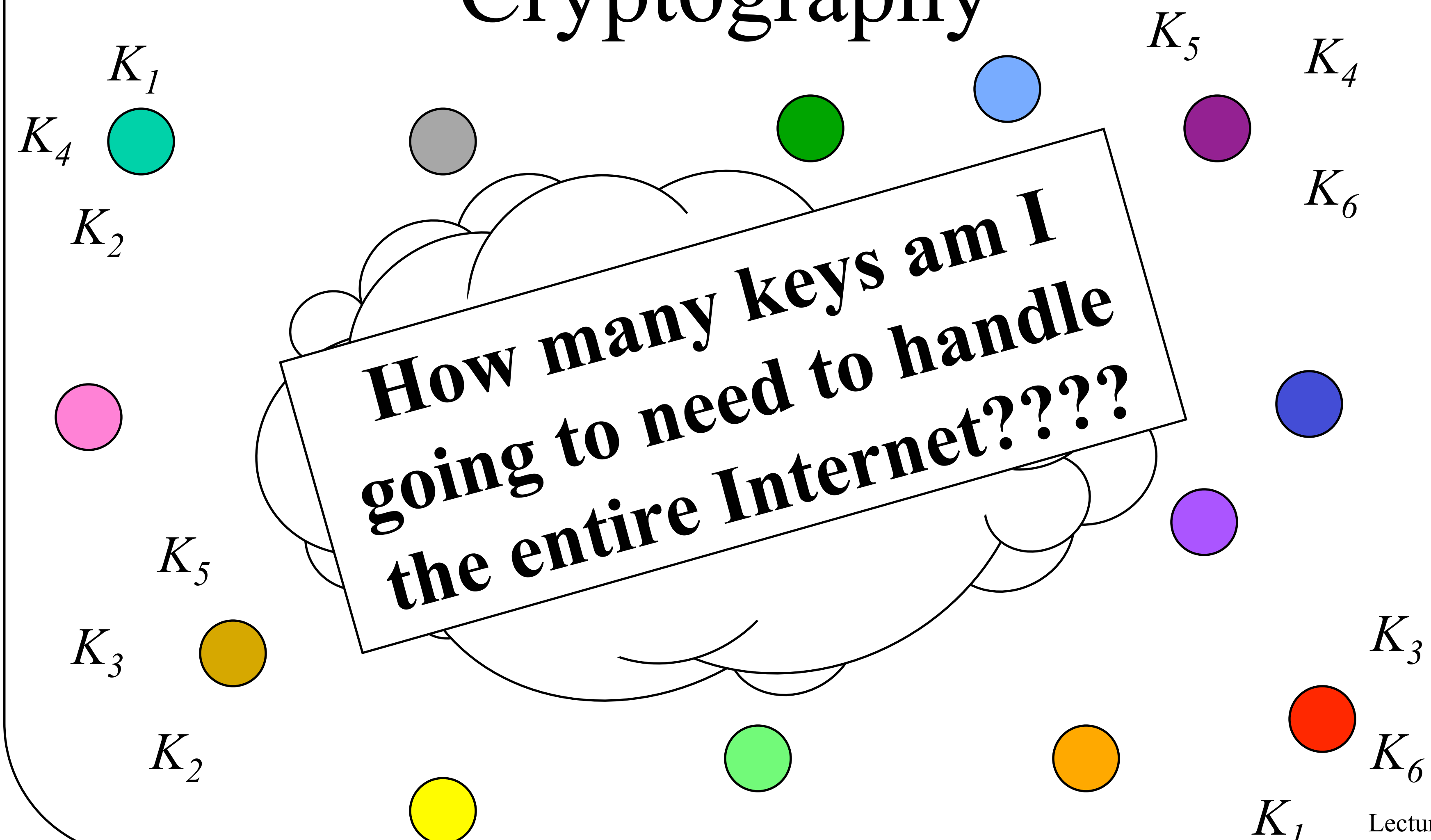
Advantages of Symmetric Key Systems

+ Encryption and authentication performed in a single operation
+ Well-known (and trusted) ones perform faster than asymmetric key systems
+ Doesn’t require any centralized authority
  - Though key servers help a lot

Disadvantage of Symmetric Key Systems

- Encryption and authentication performed in a single operation
  - Makes signature more difficult
- Non-repudiation hard without servers
- Key distribution can be a problem
- Scaling

Scaling Problems of Symmetric Cryptography



Sample Symmetric Key Ciphers

- The Data Encryption Standard
- The Advanced Encryption Standard
- There are many others

The Data Encryption Standard

- Probably the best known symmetric key cryptosystem
- Developed in 1977
- Still much used
  – Which implies breaking it isn't trivial
- But showing its age

History of DES

- Created in response to National Bureau of Standards studies
- Developed by IBM
- Analyzed , altered, and approved by the National Security Agency
- Adopted as a federal standard
- One of the most widely used encryption algorithms

Overview of DES Algorithm

- A block encryption algorithm
  - 64 bit blocks
- Uses substitution and permutation
  - Repeated applications
    - 16 cycles worth
- 64 bit key
  - Only 56 bits really used, though

More On DES Algorithm

- Uses substitutions to provide confusion
  - To hide the set of characters sent
- Uses transpositions to provide diffusion
  - To spread the effects of one plaintext bit into other bits
- Uses only standard arithmetic and logic functions and table lookup
- Performs 16 rounds of substitutions and permutations
  - Involving the key in each round

Decrypting DES

- For DES, *D()* is the same as *E()*
- You decrypt with exactly the same algorithm
- If you feed ciphertext and the same key into DES, the original plaintext pops out

Is DES Secure?

- Apparently, reasonably
- NSA alterations believed to have increased security against differential cryptanalysis
- Some keys are known to be weak with DES
  - So good implementations reject them
- To date, only brute force attacks have publicly cracked DES

Key Length and DES

- Easiest brute force attack is to try all keys
  - Looking for a meaningful output
- Cost of attack proportional to number of possible keys
- Is 2^{56} enough keys?
- Not if you seriously care
  - Cracked via brute force in 1998
  - Took lots of computers and time
  - But computers keep getting faster . . .

Does This Mean DES is Unsafe?

- Depends on what you use it for
- Takes lots of compute power to crack
- On the other hand, computers will continue to get faster
- And motivated opponents can harness vast resources
- Increasingly being replaced by AES

The Advanced Encryption Standard

- A relatively new cryptographic algorithm
- Intended to be the replacement for DES
- Chosen by NIST
  - Through an open competition
- Chosen cipher was originally called Rijndael
  - Developed by Dutch researchers
  - Uses combination of permutation and substitution

Increased Popularity of AES

- Gradually replacing DES
  - As was intended
- Various RFCs describe using AES in IPSEC
- FreeS/WAN IPSEC (for Linux) includes AES
- Some commercial VPNs use AES
- Various Windows AES products available
  - Used for at least some purposes in Vista

Public Key Encryption Systems

- The encrypter and decrypter have different keys

$$C = E(K_E, P)$$

$$P = D(K_D, C)$$

- Often, works the other way, too

$$C' = E(K_D, P)$$

$$P = D(K_E, C')$$

History of Public Key Cryptography

- Invented by Diffie and Hellman in 1976
- Merkle and Hellman developed Knapsack algorithm in 1978
- Rivest-Shamir-Adelman developed RSA in 1978
  - Most popular public key algorithm
- Many public key cryptography advances secretly developed by British and US government cryptographers earlier

Practical Use of Public Key Cryptography

- Keys are created in pairs
- One key is kept secret by the owner
- The other is made public to the world
- If you want to send an encrypted message to someone, encrypt with his public key
  - Only he has private key to decrypt

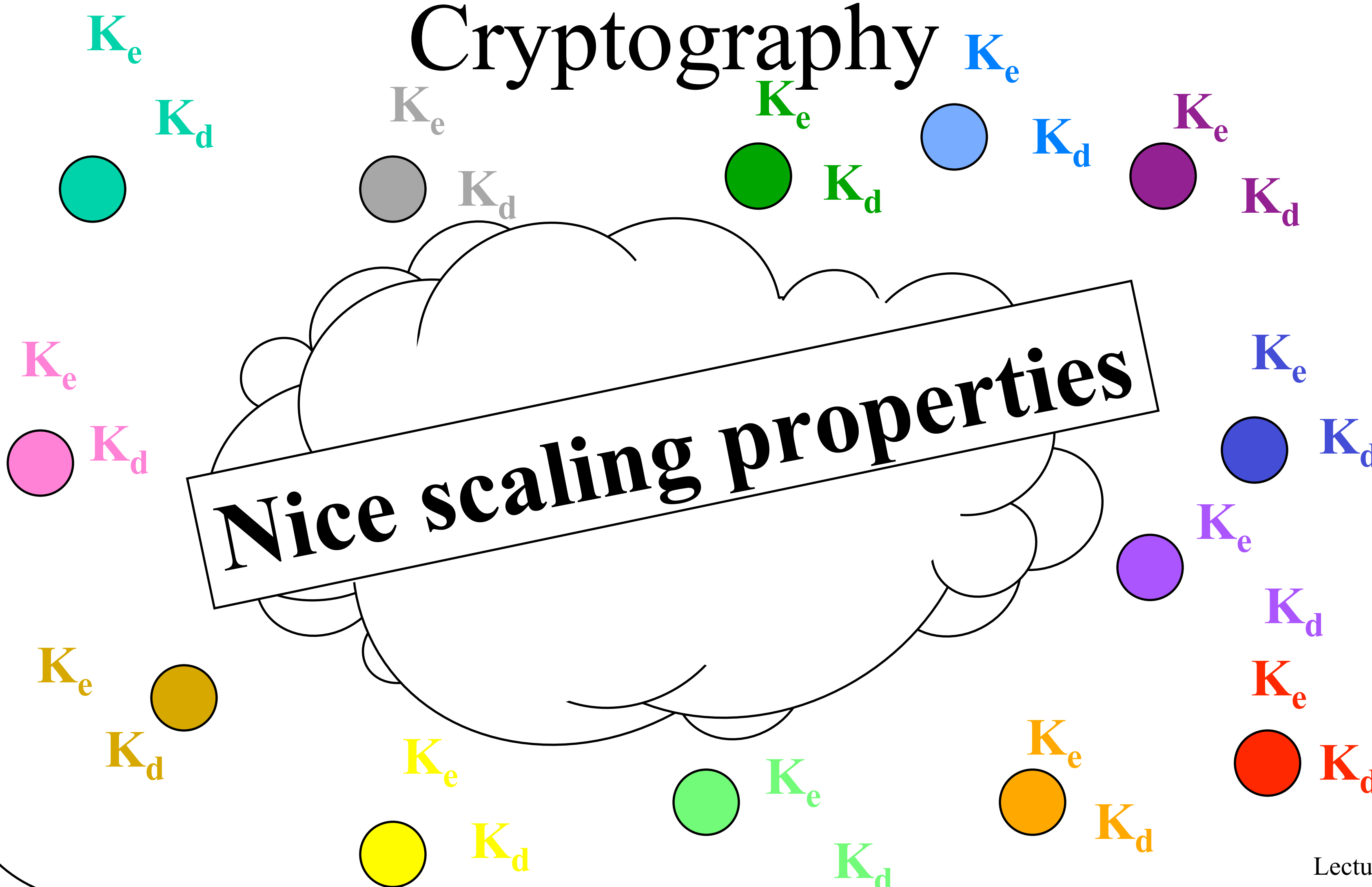
Authentication With Shared Keys

- If only two people know the key, and I didn't create a properly encrypted message -
  - The other guy must have
- But what if he claims he didn't?
- Or what if there are more than two?
- Requires authentication servers

Authentication With Public Keys

- If I want to "sign" a message, encrypt it with my private key
- Only I know private key, so no one else could create that message
- Everyone knows my public key, so everyone can check my claim directly

Scaling of Public Key Cryptography



Key Management Issues

- To communicate via shared key cryptography, key must be distributed
  - In trusted fashion
- To communicate via public key cryptography, need to find out each other's public key
  - "Simply publish public keys"

Issues of Key Publication

- Security of public key cryptography depends on using the right public key
- If I am fooled into using the wrong one, that key's owner reads my message
- Need high assurance that a given key belongs to a particular person
- Which requires a *key distribution infrastructure*

RSA Algorithm

- Most popular public key cryptographic algorithm
- In wide use
- Has withstood much cryptanalysis
- Based on hard problem of factoring large numbers

RSA Keys

- Keys are functions of a pair of 100-200 digit prime numbers
- Relationship between public and private key is complex
- Recovering plaintext without private key (even knowing public key) is supposedly equivalent to factoring product of the prime numbers

Comparison of DES and RSA

- DES is much more complex
- However, DES uses only simple arithmetic, logic, and table lookup
- RSA uses exponentiation to large powers
  - Computationally 1000 times more expensive in hardware, 100 times in software
- Key selection also more expensive

Security of RSA

- Conjectured that security depends on factoring large numbers
  - But never proven
  - Some variants proven equivalent to factoring problem
- Probably the conjecture is correct

Attacks on Factoring RSA Keys

- In 2005, a 640 bit RSA key was successfully factored
  - Took 30 CPU years of 2.2 GHz machines
  - 5 months calendar time
- Research on integer factorization suggests keys up to 2048 bits may be insecure
- Size will keep increasing
- The longer the key, the more expensive the encryption and decryption

Combined Use of Symmetric and Asymmetric Cryptography

- Very common to use both in a single session
- Asymmetric cryptography essentially used to “bootstrap” symmetric crypto
- Use RSA (or another PK algorithm) to authenticate and establish a *session key*
- Use DES/Triple DES/AES using session key for the rest of the transmission

Combining Symmetric and Asymmetric Crypto

Alice wants to share the key only with Bob

Bob wants to be sure it's Alice's key

Alice

K_{EA} K_{DA}

K_{EB}

Bob

K_{EB} K_{DB}

K_{EA}

Only Bob can decrypt it

Only Alice could have created it

K_S

$C=E(K_S,K_{EB})$

$M=E(C,K_{DA})$

$K_S=D(C,K_{DB})$

$C=D(M,K_{EA})$

Digital Signature Algorithms

- In some cases, secrecy isn't required
- But authentication is
- The data must be guaranteed to be that which was originally sent
- Especially important for data that is long-lived

Desirable Properties of Digital Signatures

- Unforgeable
- Verifiable
- Non-repudiable
- Cheap to compute and verify
- Non-reusable
- No reliance on trusted authority
- Signed document is unchangeable

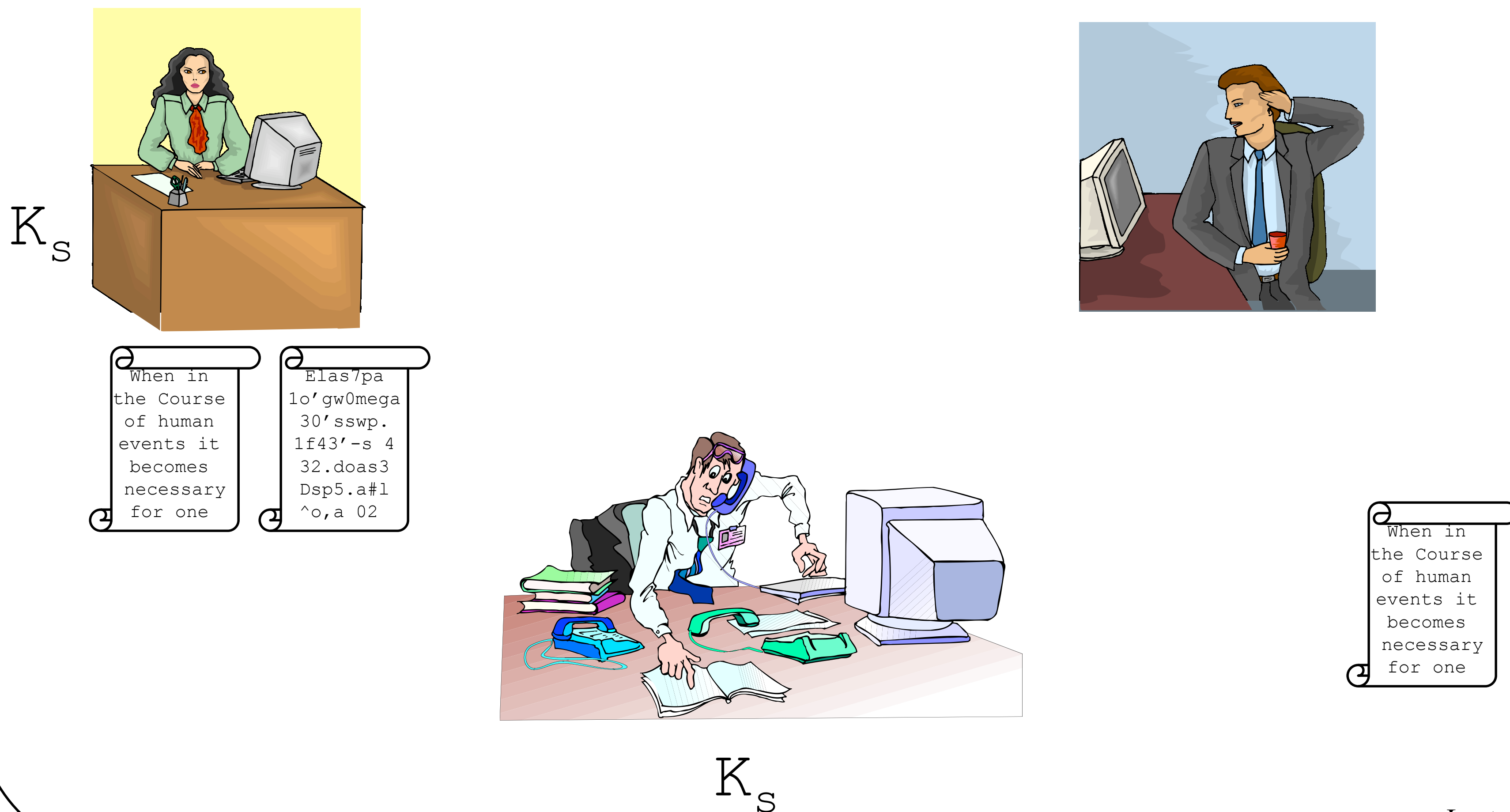
Encryption and Digital Signatures

- Digital signature methods are based on encryption
- The basic act of having performed encryption can be used as a signature
  - If only I know K, then $C=E(P,K)$ is a signature by me
  - But how to check it?

Signatures With Shared Key Encryption

- Requires a trusted third party
- Signer encrypts document with secret key shared with third party
- Receiver checks validity of signature by consulting with trusted third party
- Third party required so receiver can't forge the signature

For Example,


K
S
When in the Course of human events it becomes necessary for one
Elas7pa 1o'gw0mega 30'sswp. 1f43'-s 4 32.doas3 Dsp5.a#1 ^o,a 02
When in the Course of human events it becomes necessary for one
K
S

Signatures With Public Key Cryptography

- Signer encrypts document with his private key
- Receiver checks validity by decrypting with signer's public key
- Only signer has the private key
  – So no trusted third party required
- But receiver must be certain that he has the right public key

For Example,

K_e

When in
the Course
of human
events it
becomes
necessary
for one

Elas7pa
1o’gw0mega
30’sswp.
1f43’-s 4
32.doas3
Dsp5.a#l
^o,a 02

When in
the Course
of human
events it
becomes
necessary
for one

K_d

R[ds7 5
1sapG5(2l
1l>wcwom
#0swlts a(
GOwW03,
Whyoec4;s
*3;d0swe

Problems With Simple Encryption Approach

- Computationally expensive
  - Especially with public key approach
- Document is encrypted
  - Must be decrypted for use
  - If in regular use, must store encrypted and decrypted versions

Secure Hash Algorithms

- A method of protecting data from modification
- Doesn't actually prevent modification
- But gives strong evidence that modification did or didn't occur
- Typically used with digital signatures

Idea Behind Secure Hashes

- Apply a one-way cryptographic function to data in question
- Producing a much shorter result
- Attach the cryptographic hash to the data before sending
- When necessary, repeat the function on the data and compare to the hash value

Secure Hash Algorithm (SHA)

- Endorsed by NIST
- Reduces input data of up to 2^{64} bits to 160 bit digest
- Doesn’t require secret key
- Broken in 2005

What Does “Broken” Mean for SHA-1?

- A crypto hash matches a digest to a document
- It’s bad if two documents match the same digest
- It’s very bad if you can easily find a second document with a matching hash
- The crypto break finds matching hashes in 2^{63} operations

How Bad Is That?

- We can do things in 2^{63} operations
  - Though it's not trivial
- But the second "document" might be junk
- So relevant if that is a reasonable attack
- NIST isn't panicking
  - But is recommending phasing out SHA-1 by 2010
  - NIST just announced a competition for a new secure hash standard

Use of Cryptographic Hashes

- Must assume opponent also has hashing function
- And it doesn't use secret key
- So opponent can substitute a different message with a different hash
- How to prevent this?
- And what (if anything) would secure hashes actually be useful for?

Hashing and Signatures

- Use a digital signature algorithm to sign the hash
- But why not just sign the whole message, instead?
- Computing the hash and signing it may be faster than signing the document
- Receiver need only store document plus hash

Checking a Document With a Signed Hash

1. The party of the first part will hereafter be referred to as the party of the first part.
2. The party of the second part will hereafter be referred to as the party of the second part.

. . .

1000. The sanity clause.

K_s

Hash

111010100100110101 01

100010100 . . .

Encrypt

1. The party of the first part will hereafter be referred to as the party of the first part.
2. The party of the second part will hereafter be referred to as the party of the second part.

. . .

1000. The sanity clause.

K_p

Hash

0110111001010101011011

1010 . . .

Decrypt

0110111 . . . 1011011

101011110 . . .

MATCH!

The Birthday Attack

- How many people must be in a room for the chances to be greater than even that two of them share a birthday?
- Answer is 23
- The same principle can be used to attack hash algorithms

Using the Birthday Attack on Hashes

- For a given document, find a different document that has the effect you want
- Trivially alter the second document so that it hashes to the same value as the target document
  - Using an exhaustive attack

How Hard Is the Birthday Attack?

- Depends on the length of the hash
  – And the quality of the hashing algorithm
- Essentially, looking for hashing collisions
- So long hashes are good
  – SHA produces 2^{80} random hashes
  – But 2005 attack finds collisions in 2^{63} operations
  – Not for chosen plaintext, however