Intruders: What Can They Do?

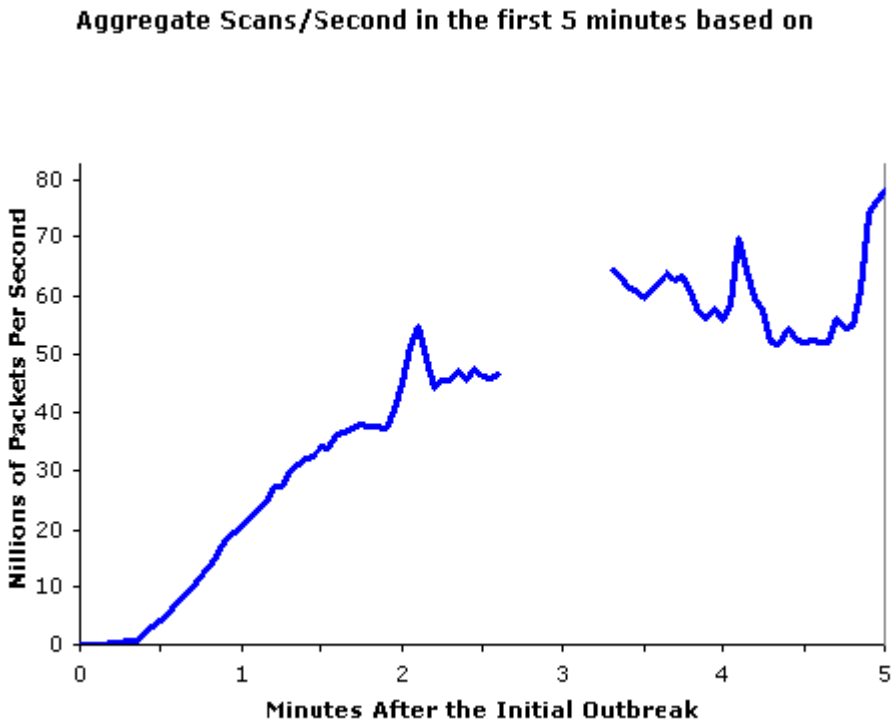
- Eavesdrop (related to confidentiality)
- Related to authentication
  - Send Messages
  - Impersonate an address and lie in wait: mutual authentication
  - Replay recorded messages
- Modify messages in transit (related to integrity)
- Write malicious code and trick people into running it

Digital Pests

- Trojan horse
  - Hidden part of some otherwise useful software
  - Today often on a Web page (Active-X, plugin)
- Virus
  - infection by receiving object (e.g., e-mail attachment), actively executing
  - self-replicating: propagate itself to other hosts, users
  - Denial of Service&DDoS:
  - Flooding Attack: Traffic analysis.
  - Energy depletion attack.

- Worm:
  - infection by passively receiving object that gets itself executed
  - self- replicating: propagates to other hosts, users

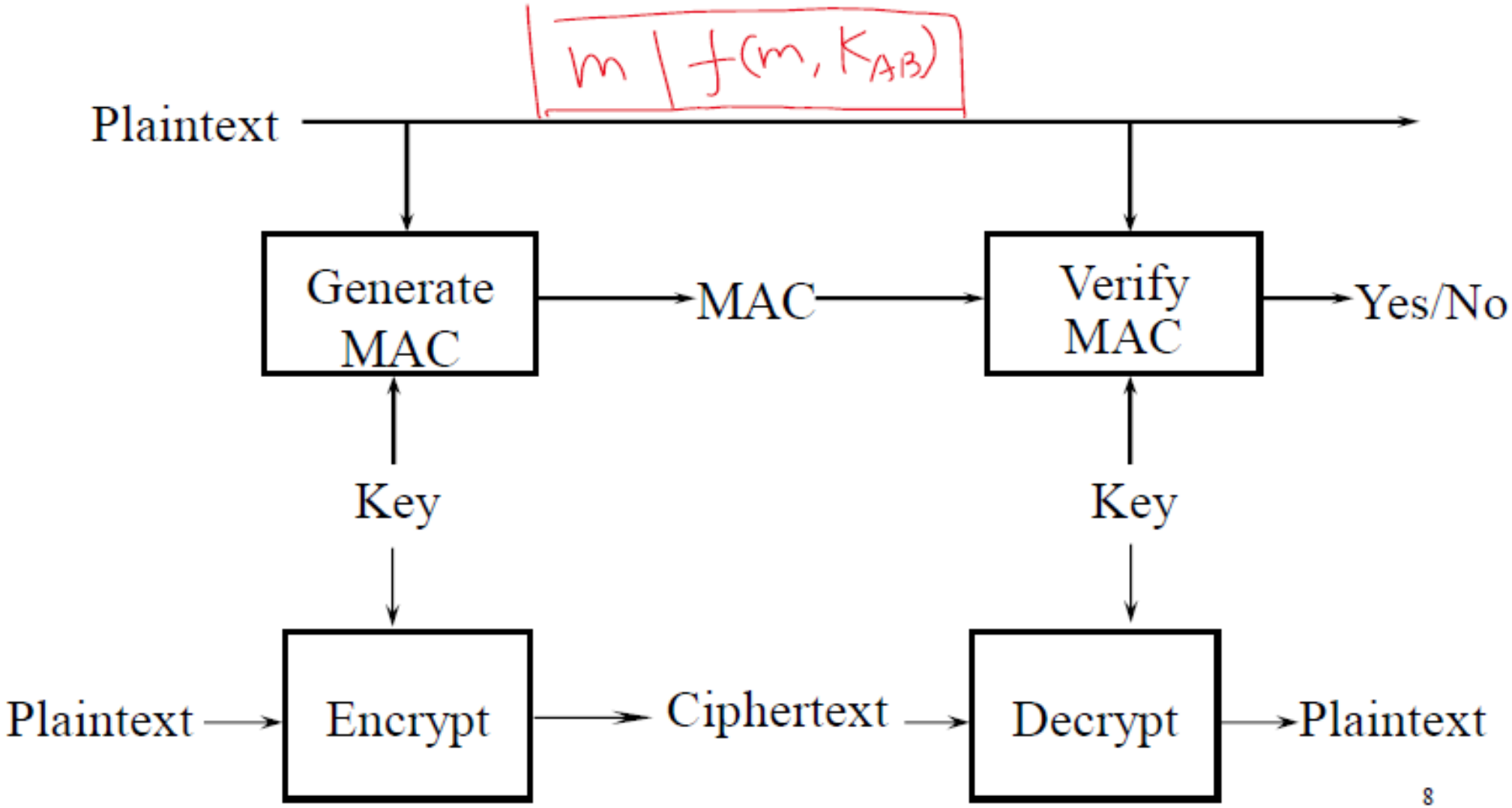
Sapphire Worm: aggregate scans/sec in first 5 minutes of outbreak (CAIDA, UWisc data)



Cryptography

- Secret writing
  - Algorithm (normally public)
  - Key: a secret value
- Three kinds of cryptographic algorithms
  - Secret Key Cryptography (DES, IDEA, RCx, AES): [Encode a message using a secret "key"] block cipher, stream cipher
  - Public Key Cryptography (RSA, Diffie-Hellman, DSS)
  - Message Digests (MD4, MD5, SHA-1)

Secret Key Integrity Protection



How Secure is Encryption?

- Depend on computational difficulty
  - An attacker who knows the algorithm we're using could try all possible keys
  - Security of cryptography depends on the limited computational power of the attacker
  - A fairly small key (e.g. 64 bits) represents a formidable challenge to the attacker: nowadays require 128bits
  - Algorithms can also have weaknesses, independent of key size
- The algorithm should be efficient for good guys to compute

How secure an algorithm is?

- Depend on computational difficulty
- A problem of mathematics: it is very hard to prove a problem is hard
- It's never impossible to break a cryptographic algorithm - we want it to be as hard as trying all keys (brutal force)
- Fundamental Tenet of Cryptography: *If lots of smart people have failed to solve a problem then it probably won't be solved* (soon)

To Publish or Not to Publish

- If the good guys break your algorithm, you'll hear about it
- If you publish your algorithm, the good guys provide free consulting by trying to crack it
- The bad guys will learn your algorithm anyway: bad guy may enter the system initially as a good guy to collect info.
- Today, most commercial algorithms are published; most military algorithms are not

Block Cipher vs Stream Cipher

- Block Cipher
  - Deterministic algorithm over fixed length block
  - Performance well studied
  - Used as building blocks in other cryptographic algorithms (e.g., universal hash, stream cipher)
- Stream Cipher
  - Typically lower hardware complexity and higher speed
  - Convenience if input length is unknown or device has limited buffer
  - Susceptible to serious security problems if used incorrectly

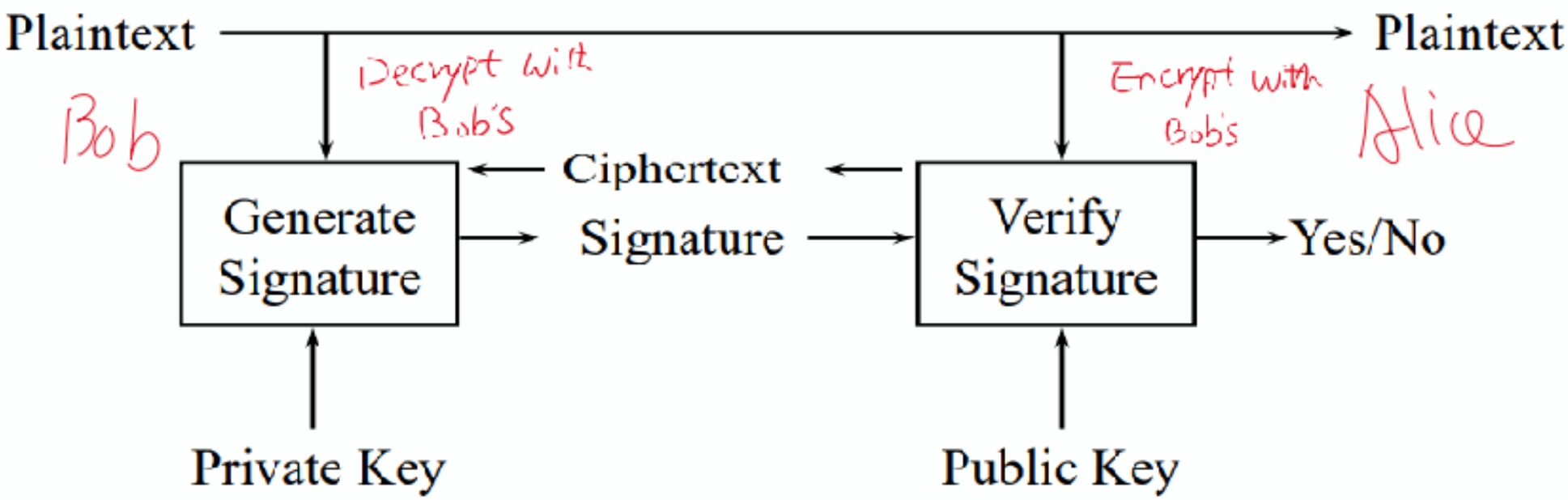
- Block cipher algorithms include DES, IDEA, AES, and Triple DES with varying key sizes and block sizes designed for efficient software implementation and secure encryption.
- Stream ciphers use bitwise operation with XOR and can be very efficient and secure when used with a "one time pad"; popular algorithms include RC2 and RC4 with variable key sizes and one-time-use keys.

Public Key Cryptography

- Two keys per user: a private key and a public key. The keys reverse each other's effects.
- Confidentiality; Digital Signature
  - When and where to use public or private key
  - Private key always links to the part to be protected
- Key distribution convenience and efficiency
  - Public key over public channel to enable secure communications
  - O(n) key distribution

Public Key: Digital Signature

- Integrity
- Signature (nonrepudiation)



Message Digest Functions

- Also known as cryptographic ***hashes***
  - Takes an arbitrary size message and mangles it into a fixed size digest
- ***Non-reversible***
  - If h(x)=y; given y, it is impossible to solve x
- ***Collision resistance***:
  - It should be impossible to find two messages with the same MD,
  - or come up with a message with a given MD
- ***Computing over the digest:***
  - Fixed length enabling protocol/standard convenience
  - Efficiency and security relying on the two properties

Authentication of People

- What you know:
  - Online password guessing (mildly unguessable)
  - Offline password guessing (verified with local calculation)
- What you have: Passive Devices, Smart Cards (NFC)
- What you are
  - Trick face ID with a picture?
  - Biometric ID stolen

Key Distribution with Firewall and IDS

- Key distribution for large numbers of users and servers can be achieved through either a Key Distribution Center (KDC) for secret keys or a Public Key Infrastructure (PKI) for public keys.
- Firewalls sit between your network and the Internet, and protect you using packet filtering and application-specific filters with rules that depend on the applications, which can be adapted by AI.
- Intrusion Detection Systems analyze network traffic and notify about any unusual activity, requiring a real art of distinguishing malicious behavior without triggering false alarms, using statistical analysis.

Authorization with ACLs Capabilities

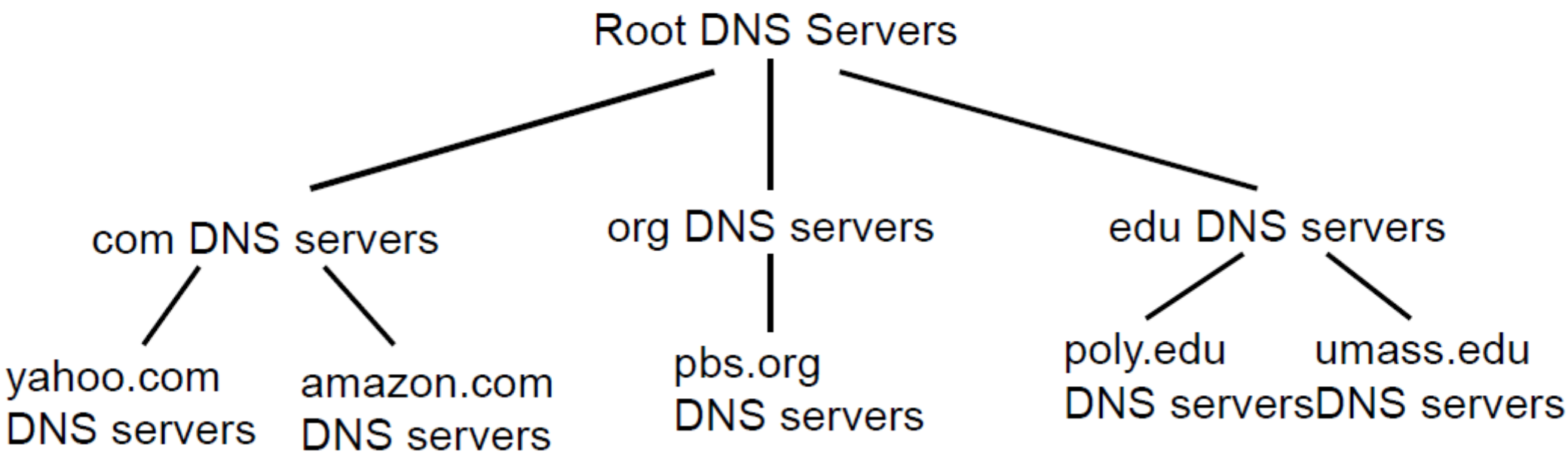
- ACLs determine access by listing authorized users/ groups, including wildcarded names and nested group membership.
- May result in slow verification and overprivileging due to coarse granularity and lack of protection within the same group.
- Authorization with Capabilities suggests using a certificate that outlines what actions are permitted, rather than identifying who the user is.
- It did not gain popularity, while current authorization methods involve servers storing membership lists of groups and KDC keeping track of user groups.

Why layering?

Dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces
  - layered reference model for discussion
  - 7 layers: physical, data link, network, transport, session, presentation, and application
- modularization eases maintenance, updating of system
  - change of implementation of layer 's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- Security in which layer? Parity Check and CRC

Domain Name System (DNS)



Client wants IP for www.amazon.com:

– client queries a root server to find com DNS server
– client queries com DNS server to get amazon.com DNS server
– client queries amazon.com DNS server to get IP address for www.amazon.com
– UDP offers best effort transmission with the possibility of lost and out-of-order packets, ideal for real-time video/audio applications, while TCP ensures packets arrive in the same order with variable and unpredictable delay, crucial for data transfer. Reliability can also be extended to information security through Transport layer security.

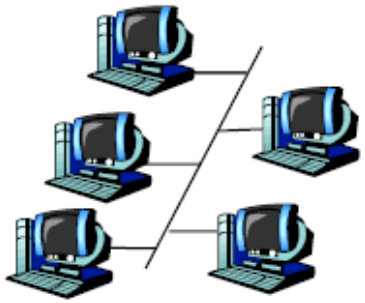
- Security issues
  - Authentication, integrity, access control, intrusion detection, DDoS

Multiple Access Links and Protocols

- Two types of "links":
- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - old-fashioned Ethernet
  - upstream HFC
  - 802.11 wireless LAN

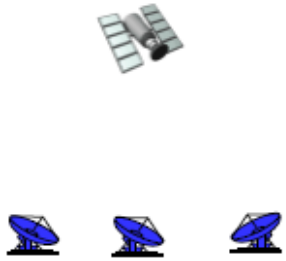
- Routers exchange messages to obtain topology and compute paths enabled by local forwarding tables, while security issues include authentication, integrity, access control, intrusion detection, and DDoS.

- Security issues
  - Traditional security issues
  - Selfish, jamming, traffic analysis



shared wire (e.g., cabled Ethernet)

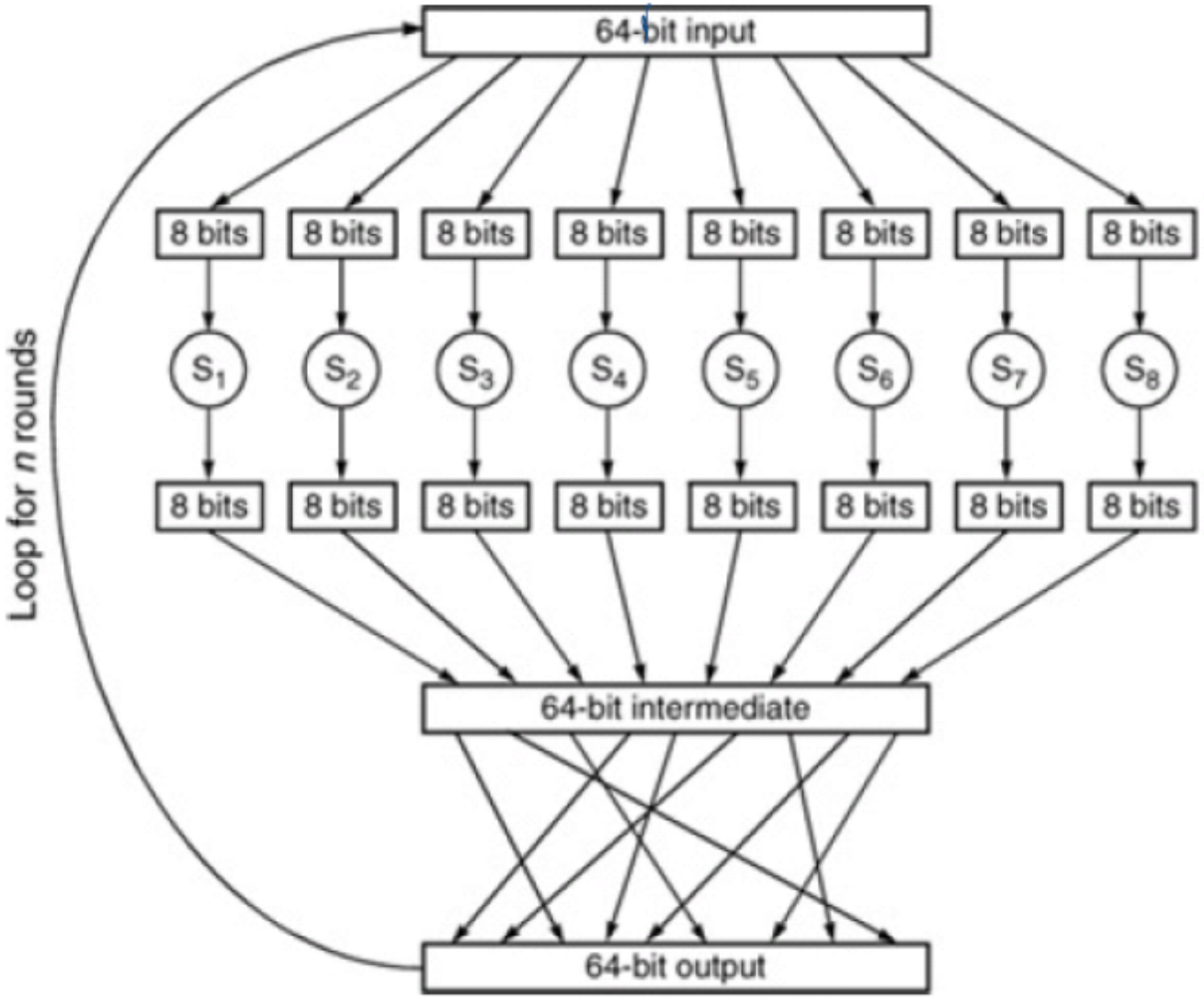
shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

humans at a cocktail party (shared air, acoustical)

Block Encryption Algorithm



Divide input into eight 8-bit pieces

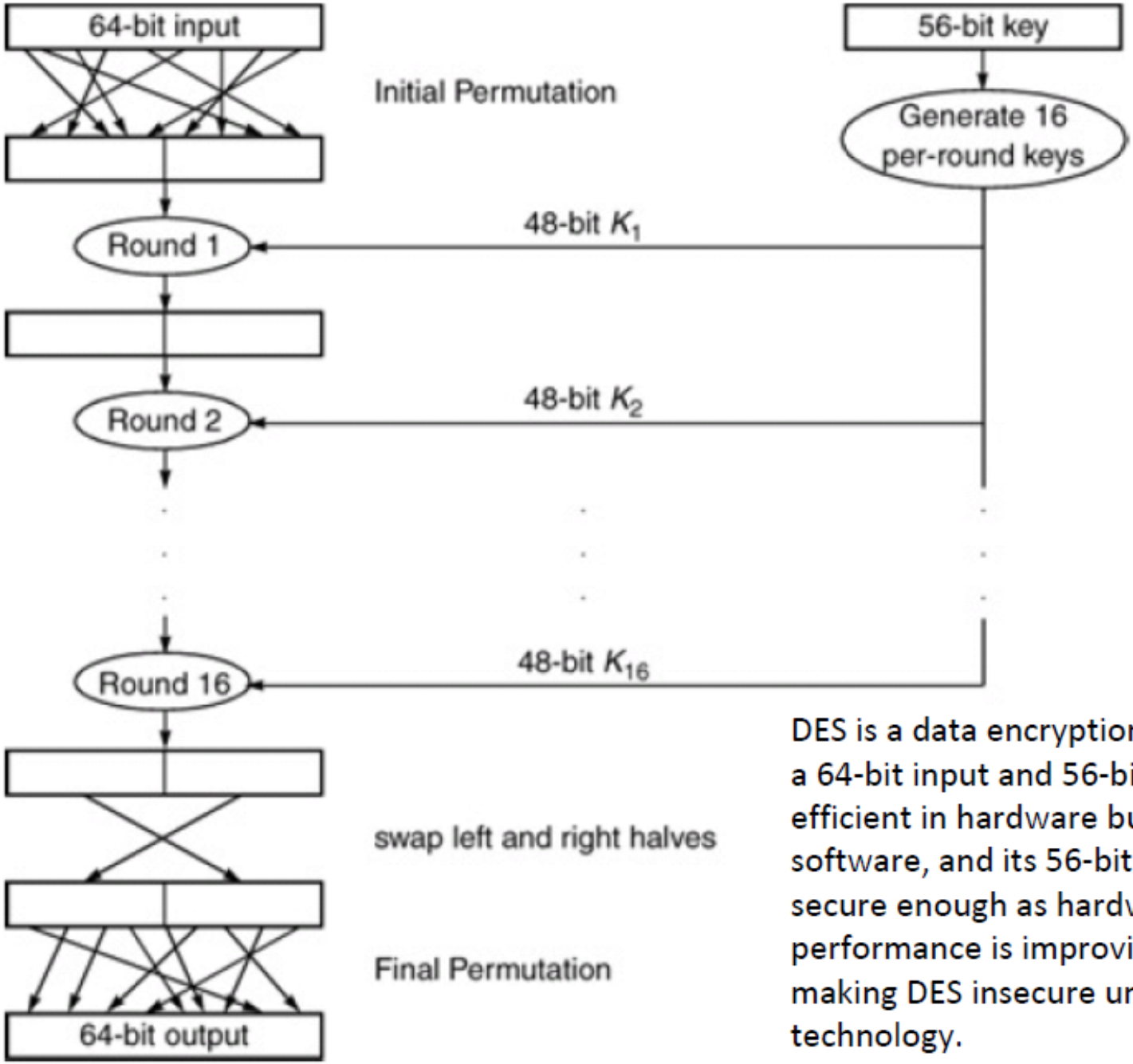
Eight 8-bit substitution functions derived from the key

Permute the bits, possibly based on the key

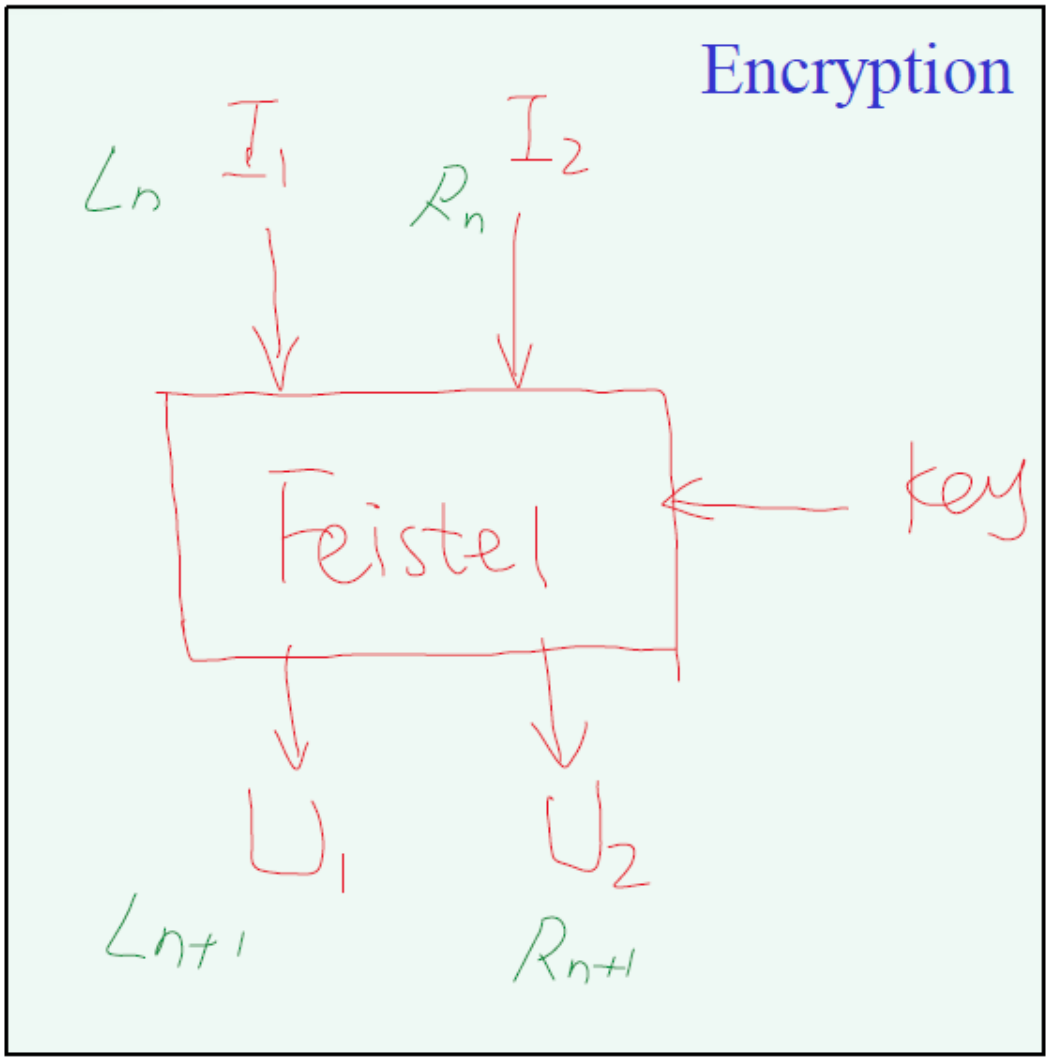
A block cipher uses a fixed-length input and key to produce a fixed-length output that is the same length as the input.

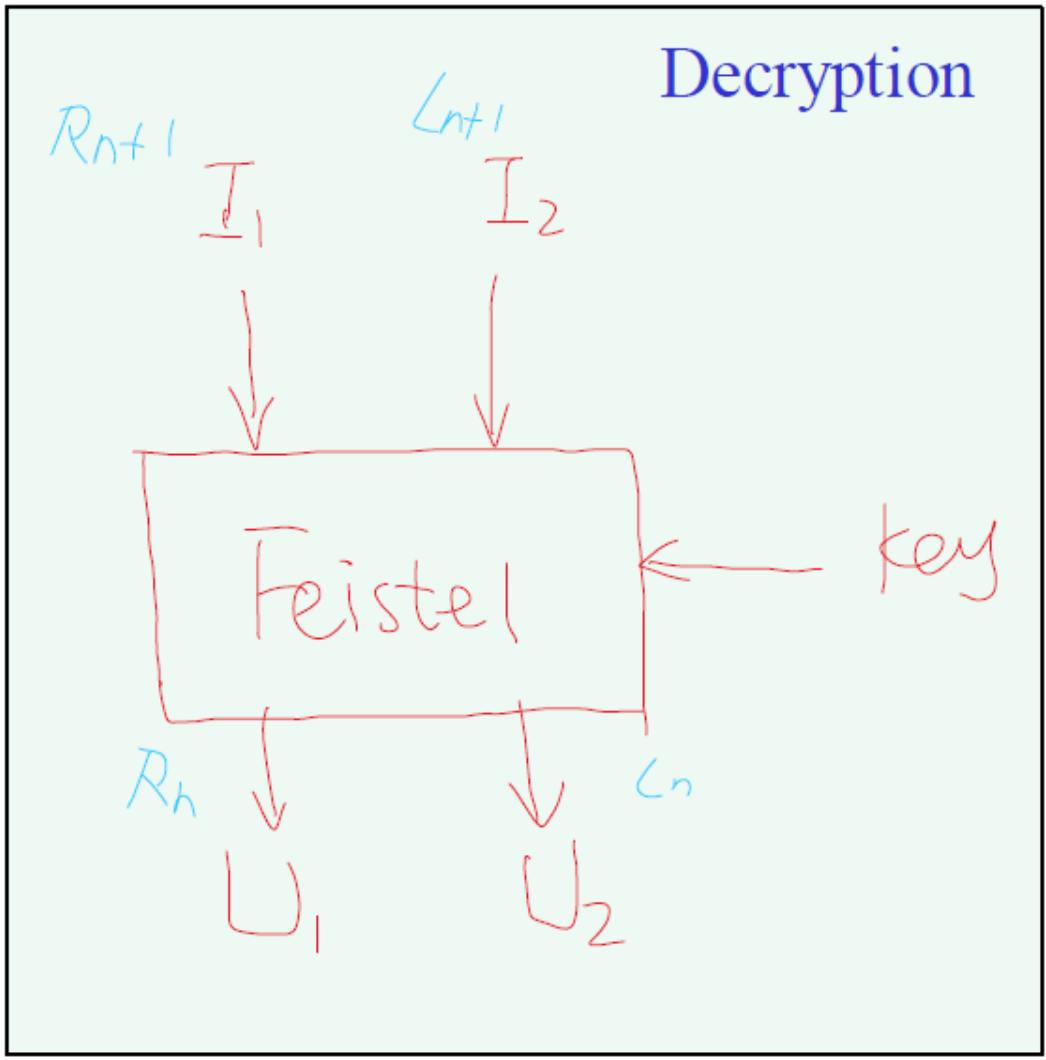
Optimal block size is crucial as small block size allows building a table, while too long block size can lead to performance penalties.

Basic Structure of DES



DES is a data encryption standard with a 64-bit input and 56-bit key that is efficient in hardware but slow in software, and its 56-bit key is not secure enough as hardware performance is improving rapidly, making DES insecure under current technology.





$U_1 = I_2$

$U_2 = I_1 \oplus M(I_2, key)$

Circuit

* Cannot reverse info. flow from the output pins

International Data Encryption Algorithm (IDEA)

- IDEA is an encryption algorithm efficient in software compute with 64-bit input, 128-bit key, and three reversible primitive operations:
  - Bitwise exclusive or ( )
  - Addition mod 2^{16} mod (+)
  - Multiplication mod $2^{16}+1$ ( )
- The Mangler Function expands a 32-bit R into 48 bits and transforms it using S-Box in eight 6-bit chunks.
- The 128-bit key is expanded to 52 16-bit keys, and each round is divided into odd and even rounds with encryption and decryption being the same, except for key expansion, where the odd round requires mathematical inverses of keys while the even round uses the same key.

Advanced Encryption Standard (AES)

- The motivation behind creating AES was the lack of security, speed, and flexibility in existing encryption standards like DES, Triple DES, and IDEA.
- AES is a standardized version of the Rijndael algorithm, developed by Joan Daemen and Vincent Rijmen, and facilitated by NIST to provide a secure, efficient, and unencumbered encryption standard. Below is Rijndael basic structure:
- The block size N_b.
  - Input is $4N_b$ octets.
  - AES has N_b=4.
- The key size N_k.
  - Key is $4N_k$ octets.
  - N_k=4, 6, 8 for AES-128, AES-192, and AES-256, respectively
- The number of rounds $N_r = 6+\max(N_b, N_k)$

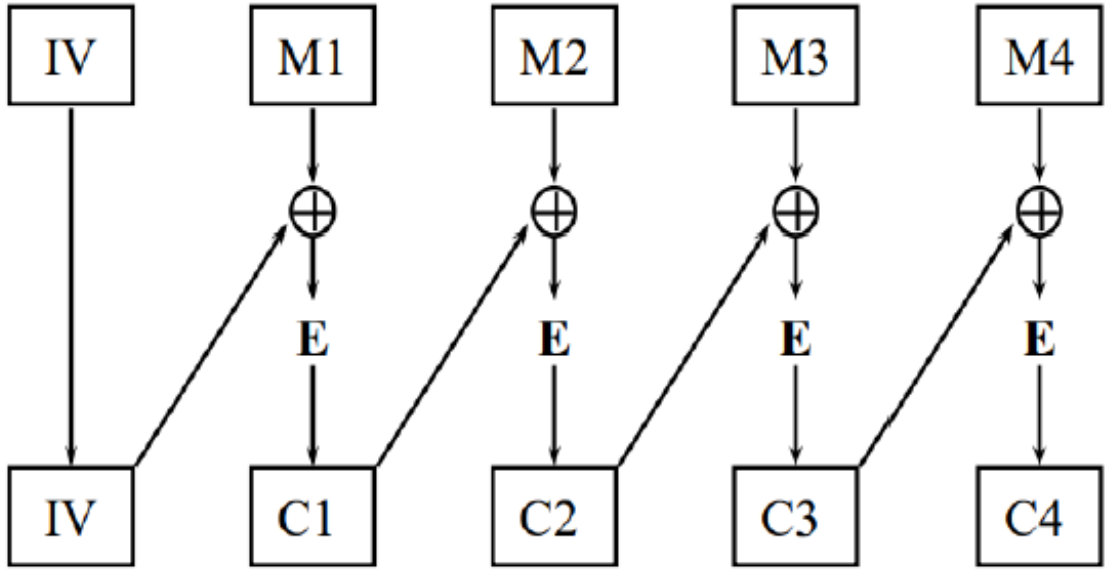
Cipher Block Chaining (CBC)

What happens if ci gets lost? Garbled? How much data gets lost?

• **Threat 1:** How can attacker that sees and can modify the ciphertext, and knows the plaintext, modify the plaintext in a predictable way? What other effects will it have? (Known plaintext attack)

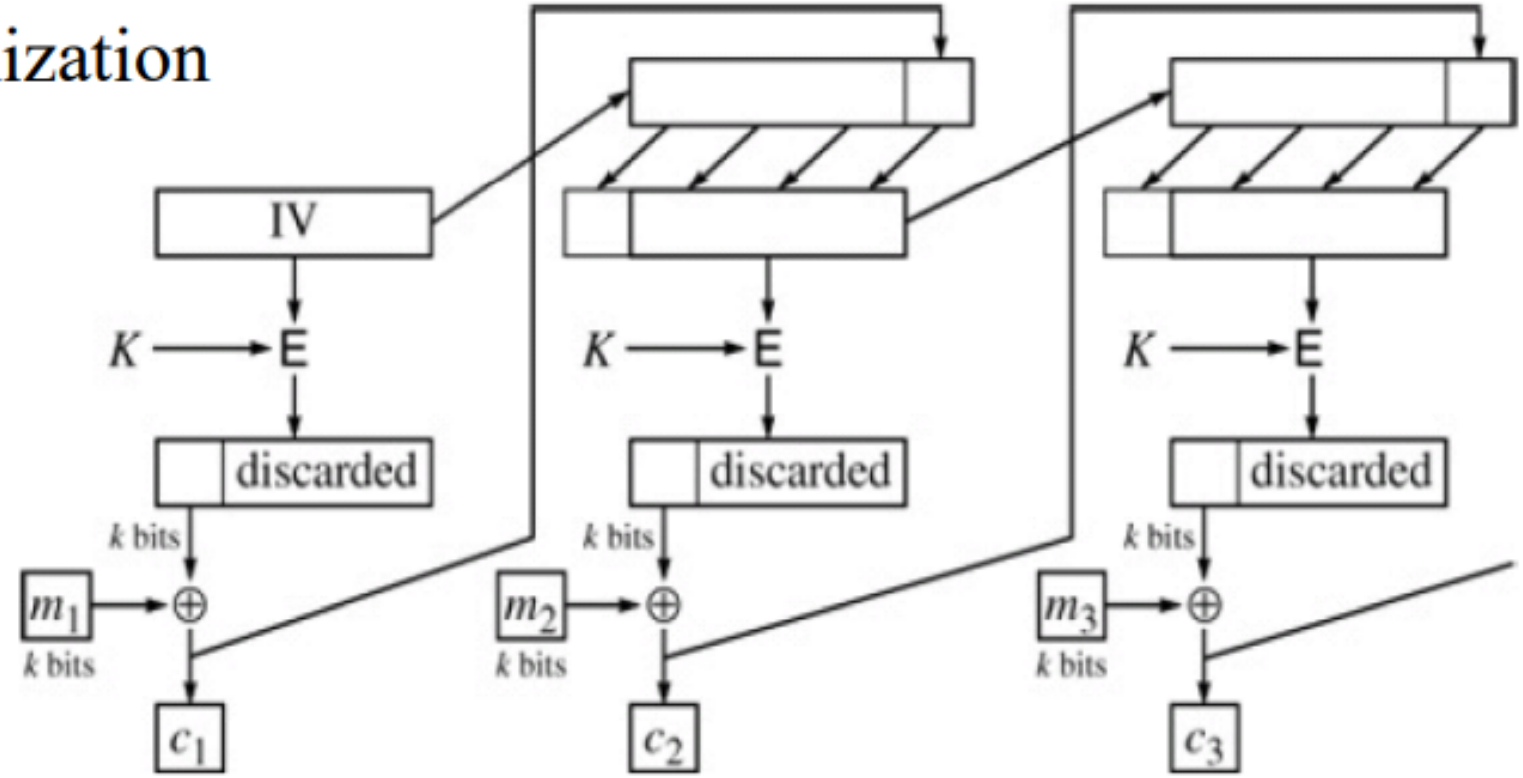
– Attach the plain text with a CRC, then do the CBC (integrity + confidentiality)

• **Threat 2:** Rearranging blocks to break the CRC protection (textbook 4.2.2.2)



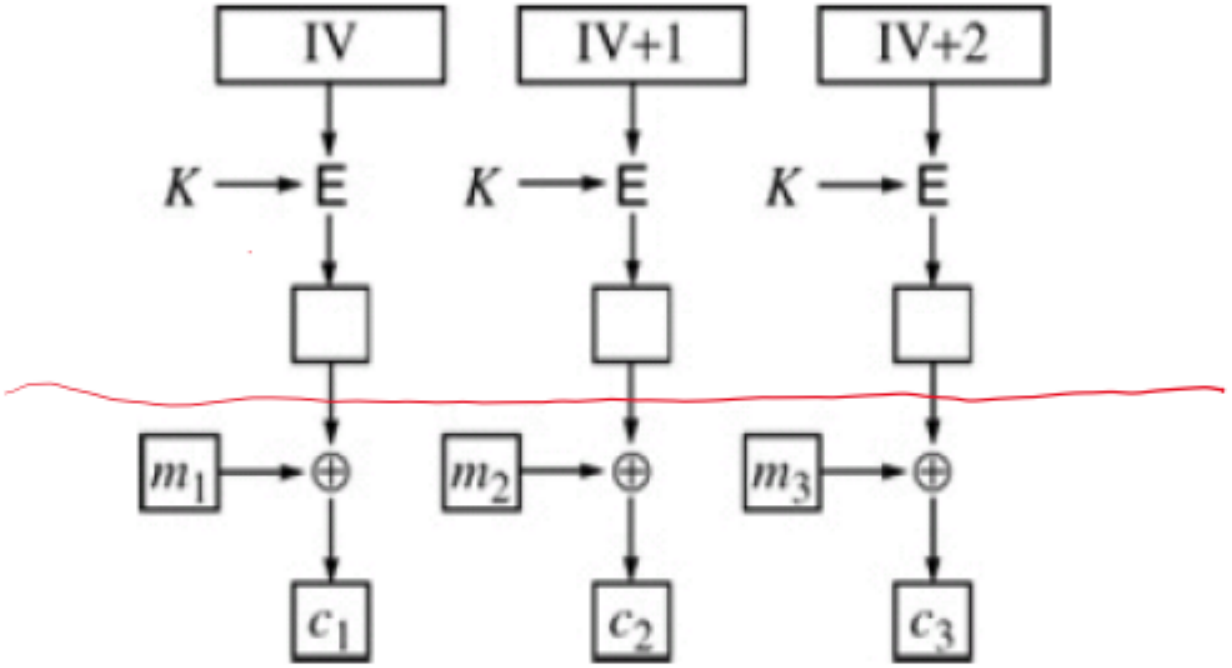
Cipher Feedback Mode (CFB)

- K-bit CFB
  - Padding bits generation coupled with messages
  - Only suitable for online operation
  - Resynchronization



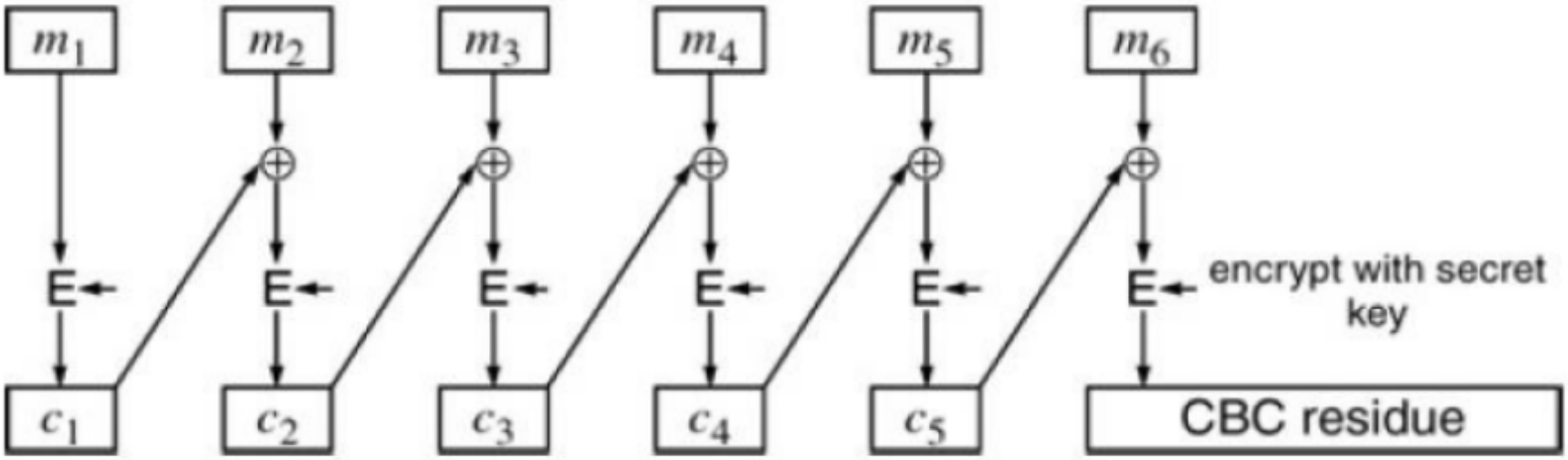
Counter Mode (CTR)

- c_i=f(key, IV, block number, p_i), pad can be calculated in advance
- Can decrypt an arbitrary block on line (useful for, e.g., random access file encryption)
- Vulnerable to known plain text attack



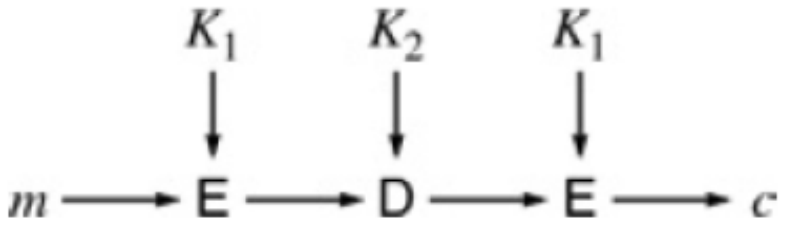
Generating MACs

- In CTR security analysis, changing the IV and key after one or more messages and attaching an index to each message can detect removing/rearranging blocks, while a modifying plaintext attack can be conducted easily.
- Encryption ≠ integrity protection
- CBC, CFB, OFB, and CTR may suffer from modifying plaintext attack, if used without integrity protection
- Cipher Block Chaining Residue (CBC residue) as MAC
- To ensure privacy and integrity together, possible solutions include CBC encryption with CRC or CBC residue, CBC encryption with plaintext and CBC residue, or CBC encryption with separate keys for encryption and integrity protection. However, it is uncertain which solution works best as each has its vulnerabilities.

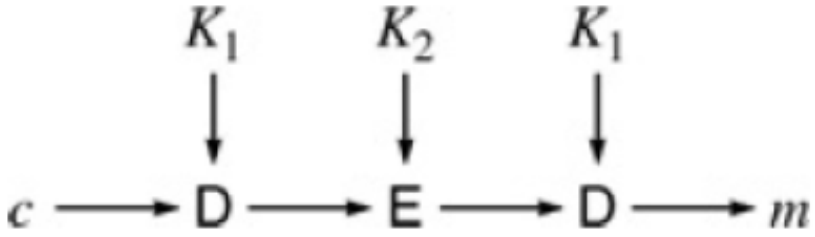


Multiple Encryption DES

- Using multiple rounds of encryption/decryption to enhance security generally applies
- 3DES (EDE)
  - 1. Two keys are used K_1 and K_2
  - 2. Encryption



Decryption

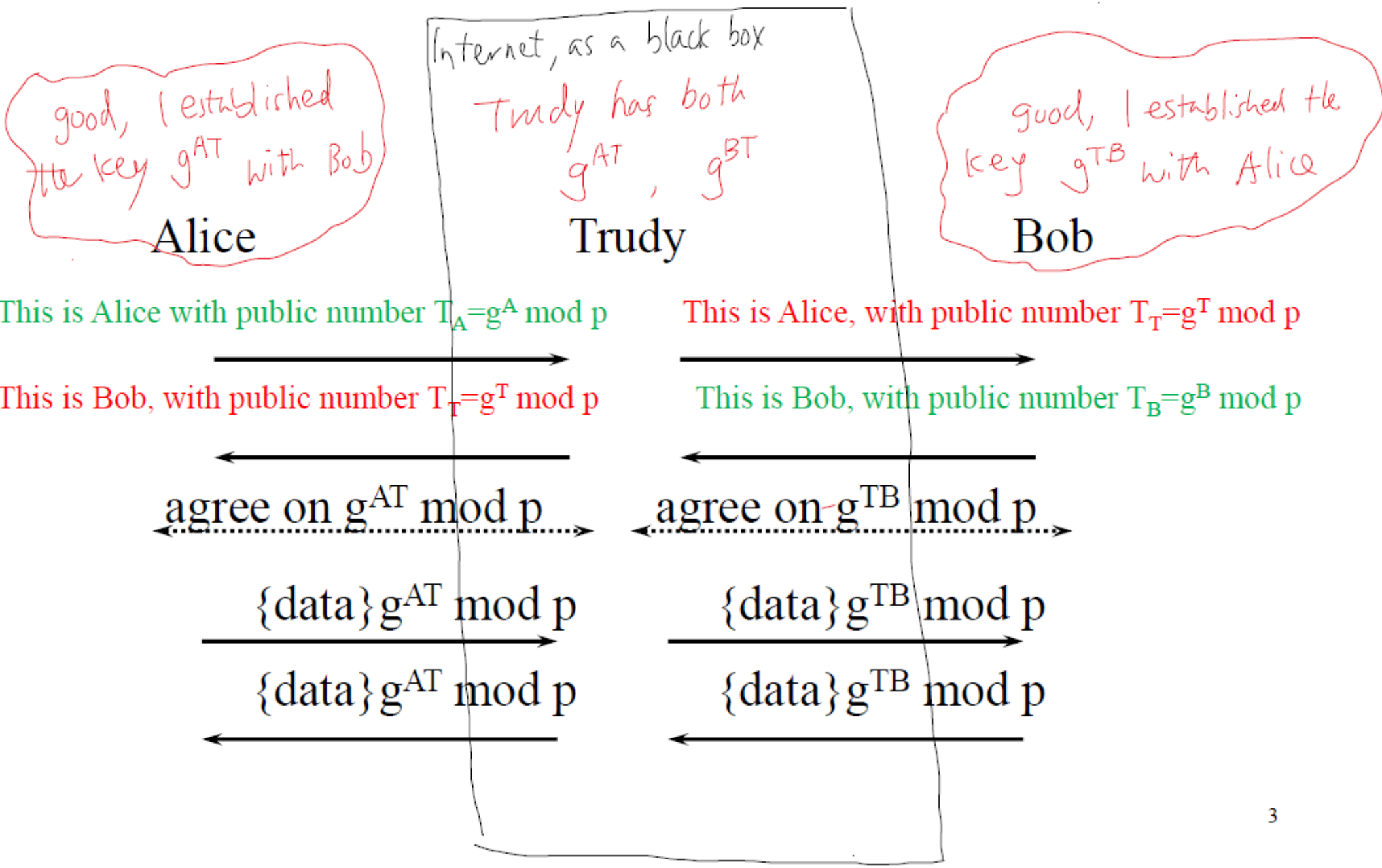


  - 3. CBC is used to encrypt large messages.

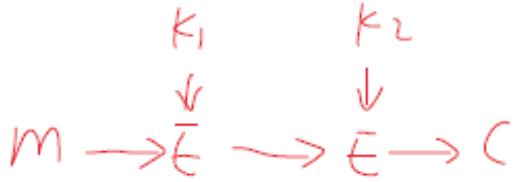
Meet in the Middle Attack

- Assume knowing a few <plaintext, ciphertext> pairs $<m_i, c_i>$
- Make table A with 2^{56} entries of $<K_A, E(m_1, K_A)>$
- Make table B with 2^{56} entries of $<K_B, D(c_1, K_B)>$
- The right keys, K_A and K_B, can be identified with a probability close to 1 by finding the entries with $E(m_1, K_A)=D(c_1, K_B)$ and matching them over three <plaintext, ciphertext> pairs.
- Meet in the Middle Analysis involves finding a pair of keys by matching entry pairs in Table-A and Table-B, where the probability of a given match is $1/2^8$ and on average, there are 2^{48} matching entry pairs.

Man in the Middle



Performance analysis



- Meet in the middle attack
  - Complexity: $O(2x2^{56}) + O(2^{48})$, that is, around $O(2^{57})$
- Triple Encryption with Only Two Keys
  - EDE with two keys has security level 112 bits
  - Using EDE with three different keys, it is manageable, with complexity $O(2x2^{32})$, to find a triple of keys that maps a given plain text to a given ciphertext
  - Setting K1=K3 increases the complexity to $O(2^{64})$.
- Why EDE instead of EEE? The initial and final permutations would cancel each other out with EEE, while EDE is compatible with single DES if K1 equals K2.
- CBC outside suffers from modifying plaintext attack, and EDE with a 112-bit key can be used as a modular design with any chaining method, with a sequential delay. vs. CBC inside allows pipeline operation, eliminates plaintext manipulation but has subtle security flaws.

Hash

- takes arbitrary sized input, generates fixed size output
- cryptographic hash
  - one-way (computationally infeasible to find input for a particular hash value)
  - collision-resistant (can't find two inputs that yield same hash)
  - output should look "random"
- Hash has various applications such as storing digests of files, irreversible password hash database, authentication, encryption, integrity protection, and the significant difference between a secret key algorithm and a hash algorithm is that hash is not reversible and does not require a key.
- To create encryption with hash, a pad is created using the IV and key (K), where each pad is hashed with the previous pad until the desired length is reached, and while Alice can precompute the pad with the IV, Bob can't until he sees it.

MACs with Hashes

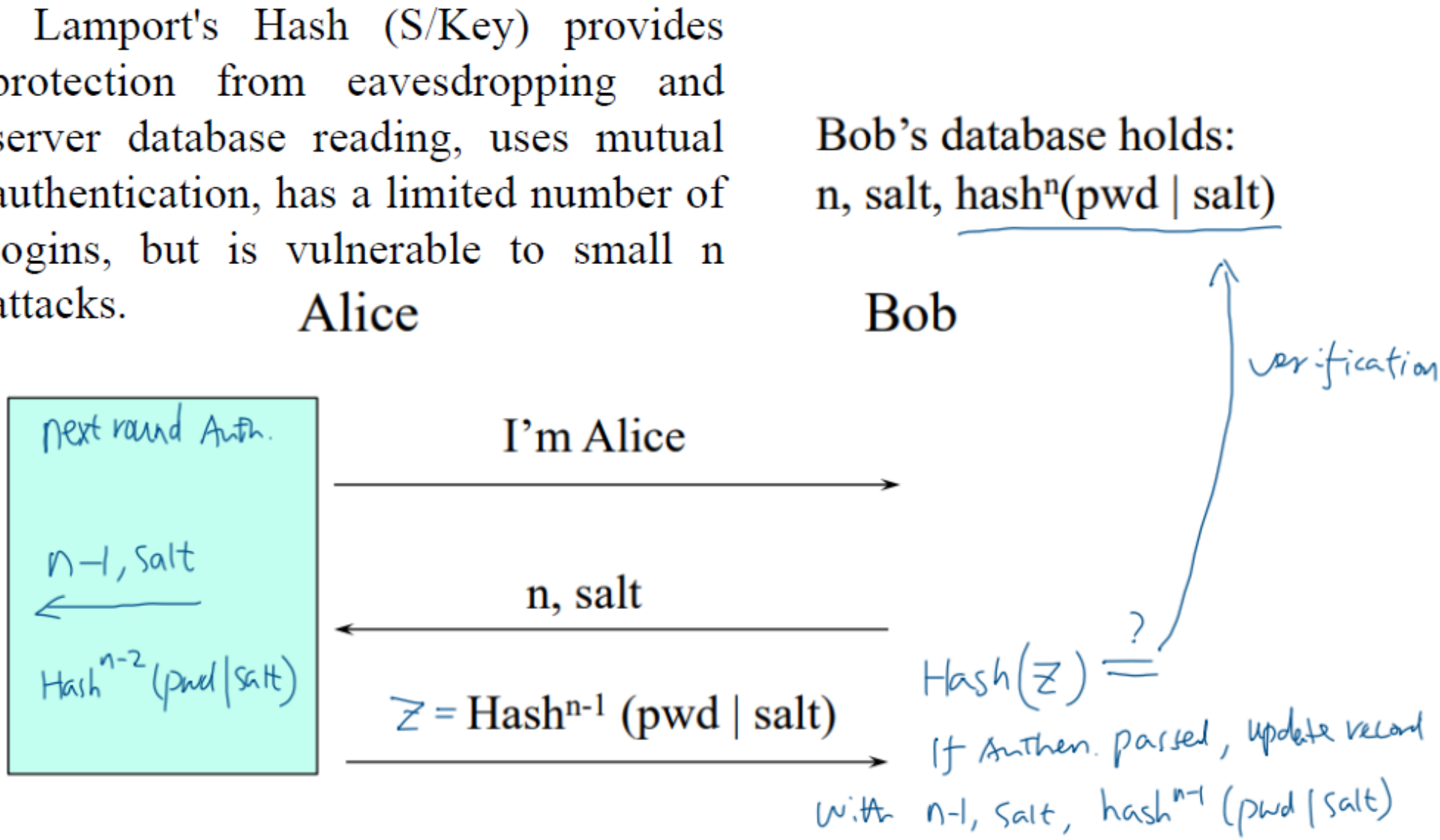
- Use MD(K_{AB}|m) as MAC
- Some (most) hash algorithms can continue from where they left off.
  - So even if you didn't know the key, if you knew m and hash(key | m) you could continue
- hash (message | key)
  - some theoretical concern
- Secure algorithms: ensure internal state to be unobservable
  - use only half the bits of the hash
  - hash(key | message | key)
  - HMAC

Some more design and analysis

- Salt
  - Store hash(pwd, salt)
  - To protect a database of hashed passwords, a non-secret salt is used that is different for each user, resulting in different hashes for users with the same password.
  - Prevents intruder from computing hash of a dictionary, and comparing against all users
- Cookie Analysis
  - Hypothesized crypt(username|server secret)
  - The cookie for usernames with the same prefix of at least 8 characters is crypt(username), and only the first 8 characters of the input should be considered.

Lamport's Hash (S/Key)

- Lamport's Hash (S/Key) provides protection from eavesdropping and server database reading, uses mutual authentication, has a limited number of logins, but is vulnerable to small n attacks.



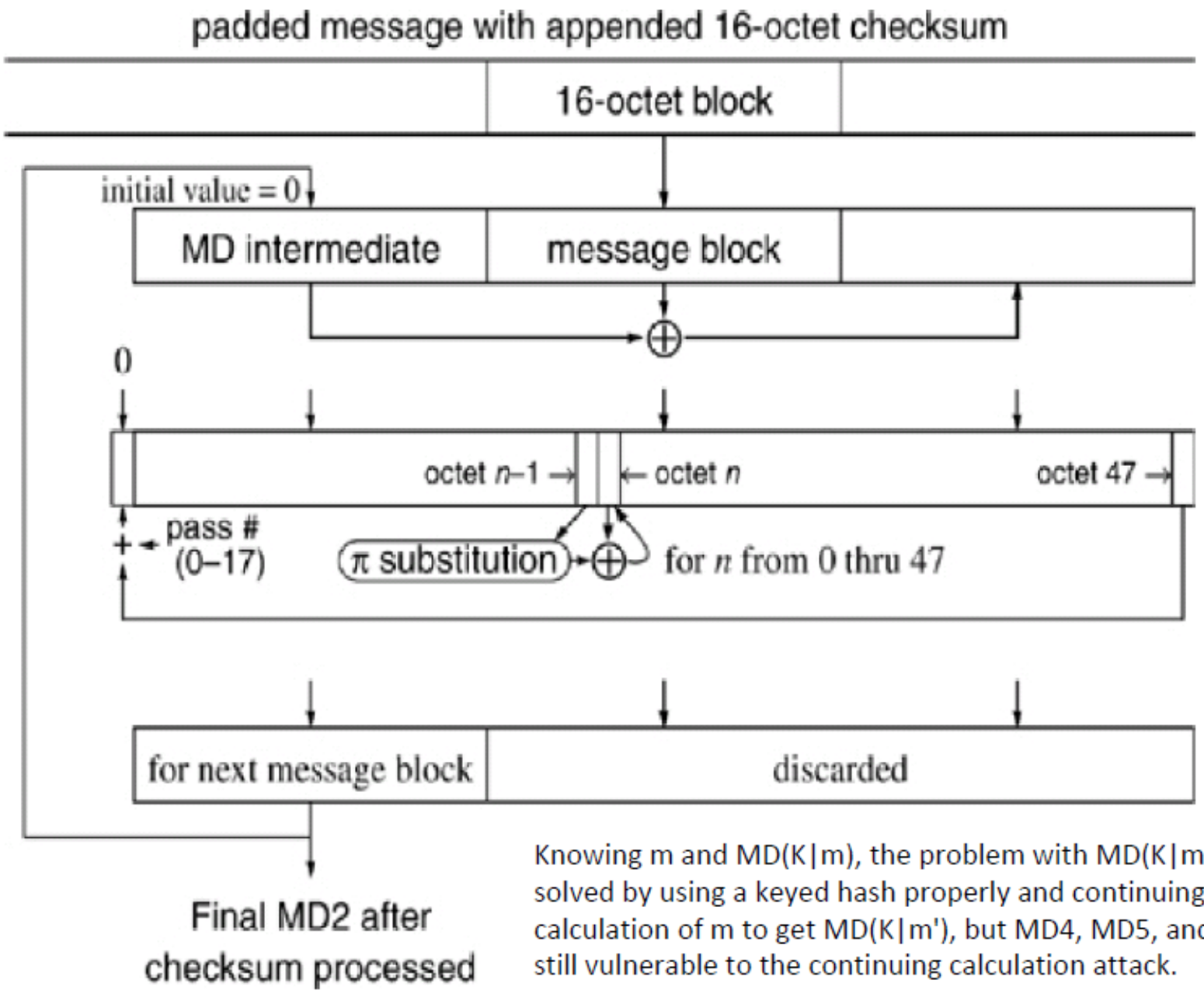
Comparing Authentication Methods

	Complexity	Robustness to server reading attack
Public key	High	high
Secret key	middle	low
Lamport's Hash	low	middle Eqv. to offline dictionary attack

Micali Revocation

- In certificate, include two quantities:
  - $hash^n$(constant1) hash(Certificate Serial number i, Server secret 1, $Salt_i$)
  - hash(constant2) hash(Certificate Serial number i, Server secret 2, $Salt_i$)
- Every "day" (...revocation interval) if cert still valid, reveal one less hash of constant1 to user. Enables user to prove not revoked
  - Refresh the status during idling period
- If revoked, reveal constant2 and post it

MD2 Final Pass



Knowing m and MD(K|m), the problem with MD(K|m) can be solved by using a keyed hash properly and continuing from the MD calculation of m to get MD(K|m'), but MD4, MD5, and SHA1 are still vulnerable to the continuing calculation attack.

A Summary of RSA

- Uses modular exponentiation
  - Choose a modulus n=pq, p, q primes
  - Totient function $\Phi(n)=(p-1)(q-1)$
  - Pick a public exponent e, compute private number by $d*e = 1 \bmod \Phi(n)$
- Public key encryption is: $\text{ciphertext} = \text{plaintext}^{e} \bmod n$
- Public key decryption is: $\text{plaintext} = \text{ciphertext}^{d} \bmod n$
- Bad guy knowing the public key (e, n)
  - Cannot figure out the factors p and q, thus cannot know $\Phi(n)$
  - Cannot figure out the private key d.

What does factoring have to do with it?

- Define $\Phi(n)$ to be the # of non-zero integers < n and relatively prime to n — ***totient function***
- Theory: $x^{y} \bmod n$ is the same as $x^{(y \bmod \Phi(n))} \bmod n$, if n is a prime or a product of 2 distinct primes
- If we can find $d*e = 1 \bmod \Phi(n)$, they'd be "exponentiative inverses"
  - (e,n) public key
  - (d,n) private key
- If $n=p*q$ (p,q primes), $\Phi(n)=(pq-p-q+1)=(p-1)(q-1)$
- Euclid's algorithm can compute a value of d such that $d*e = 1 \bmod F(n)$, but knowledge of how to factor n is required to determine F(n) and find d from e.

How do you test for primality?

- **Euler's Theorem**: For any x relative prime to n, $x^{\Phi(n)} \bmod n = 1$, or $x^{\Phi(n)} = 1 \bmod n$
- **Fermat's theorem** (note: Fermat was born 100 years earlier than Euler..it's a special case of Euler's theorem)
  if p prime and $0<x<p$, $x^{p-1} \bmod p = 1$
- So to test if n is a prime, pick x and raise x to p-1. If it's not 1, n definitely not prime
- But can it be 1 even if n not prime? Yes, but probably not. Can use different x's

Carmichael Numbers

- Annoying non-prime n's that for all x,
  $x^{n-1} \bmod n = 1$
- Luckily they are very rare
- But **Miller/Rabin test** catches these
  - n-1 = odd number c times a factor of 2: $= 2^b c$
  - choose x, raise it to c, then keep squaring (<=b)
  - If thing before 1 not = -1, not prime; if get -1, pass this test; if final result not 1, not prime
  - It uses the Fermat test to test a number "n" for primality by checking multiple values of "x", and with "k" rounds of testing, the probability of a false positive result is less than $(1/2)^k$.

Square roots of 1 mod n

- If n is prime, the square root of 1 mod n can only be 1 and -1 mod n
- So if n non-prime but a Carmichael number, at least 50% probability that thing you squared to get 1 isn't 1 or -1 mod n
  - Chinese remainder theorem
  - Construct 4 square root of 1 mod n, if n=pq

General Exponentiation

- Express the exponent in binary bits
- Start with an initial value of 1
- Read the exponent bits from high-order bit to low-order bit
  - If the bit is 1, square your value and then multiply by the base
  - If the bit is 0, just square your value
- Perform modular reduction after each operation

Optimizing RSA Public Key ops

- Turns out RSA secure even if e in (e,n) is small (like 3 or $2^{16}+1$)
- Advantages with e=3
  - Operation of encryption more efficient
  - Signature verification more efficient
  - Bring convenience to generate d
- It is not ok to use a small d

Other arcane RSA threats

- If you just encrypt a guessable plaintext, eavesdropper can verify a guess
- Trivial to forge a signature if you don't care what you're signing
  - Any number $x<n$ is a signature of $x^e \bmod n$
- Smooth numbers (sign msg combo of other messages already signed)
- If pad on right with random data,
  - Very possibly, making the number not a product of certain primes
  - someone can choose padding such that msg will be perfect cube

PKCS Padding

PKCS Standards address threats such as encrypting guessable or short messages, multiple recipients, and trivial signature forge, while also avoiding signing smooth numbers.

- Encryption

0	2	at least eight random nonzero octets	0	data

- Signing

0	1	at least eight octets of ff_{16}	0	ASN.1-encoded digest type and digest

Signed Diffie-Hellman (Avoiding Man in the Middle)

Alice | Bob

choose random A | choose random B

$[T_A = g^A \bmod p]$ signed with Alice's Private Key →

← $[T_B = g^B \bmod p]$ signed with Bob's Private Key

verify Bob's signature | verify Alice's signature

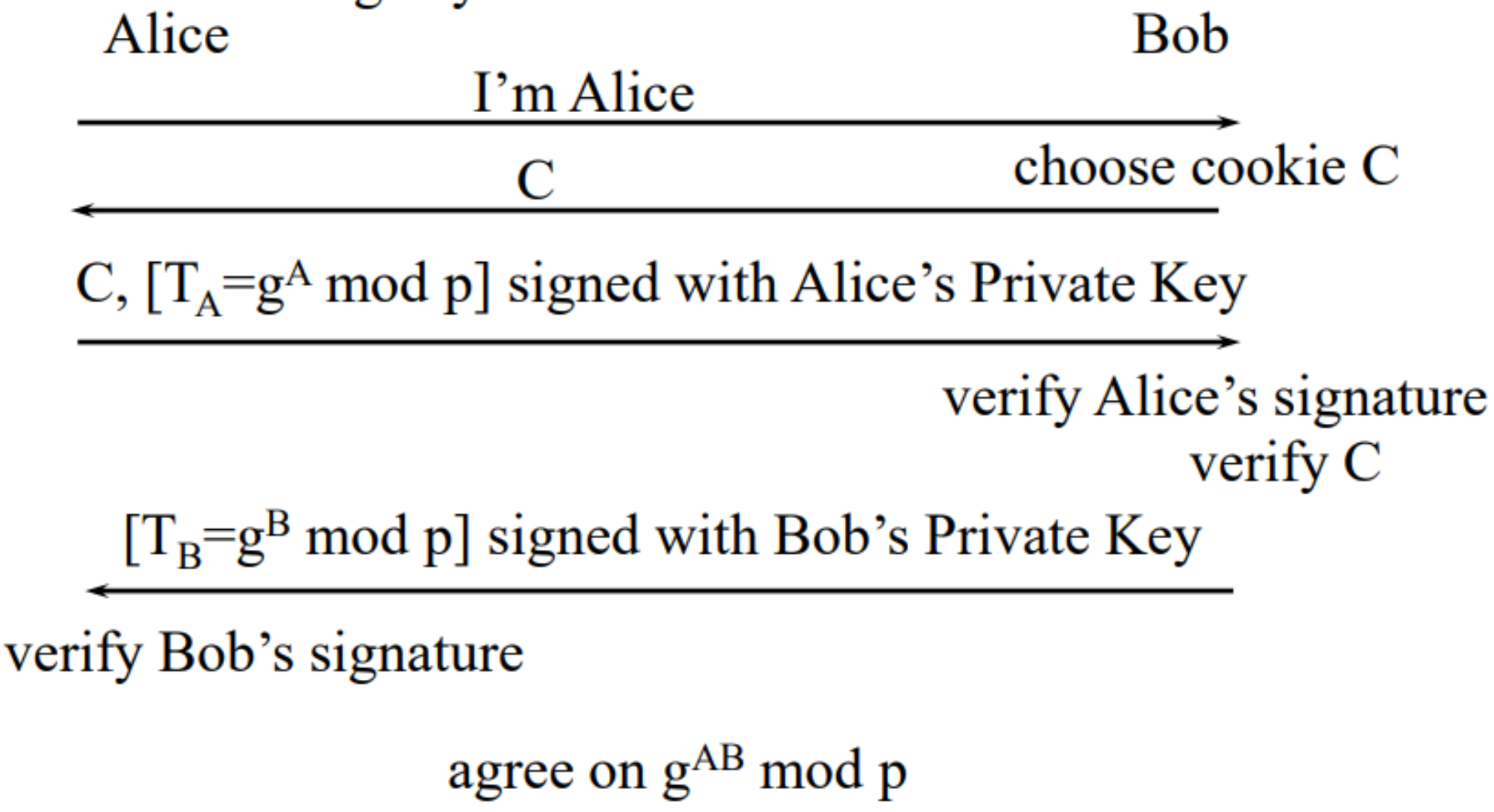
agree on $g^{AB} \bmod p$

Authenticated Diffie-Hellman

- Sign the Diffie-Hellman value with private key
- Encrypt the Diffie-Hellman value with other side's public key
- Encrypt the Diffie-Hellman exchange with the pre-shared secret
- After the D-H exchange, transmit a hash of the agreed-upon shared D-H value, your name, and the pre-shared secret
- After the D-H exchange, transmit a hash of the D-H value you transmitted and the pre-shared secret
- Diffie-Hellman (D-H) provides Perfect Forward Secrecy (PFS) and prevents third-party decryption even if they break into both parties after the conversation ends or if the private key is compromised.
- The Internet Engineering Steering Group (IESG) strongly encourages the use of PFS in protocols.

Cookie Mechanism

- The steps achieve authenticating A, authenticating B, and then sharing key



ElGamal Signatures

- Long term public/private key pair
  - Public <g, p, T>;
  - Private S with $g^S \bmod p = T$
- Per-message public/private key pair
  - Private S_m; public $T_m = g^{Sm} \bmod p$
- ElGamal Signature
  - Message digest $d_m = h(m|T_m)$
  - Signature: $X = S_m + d_m S \bmod (p-1)$
  - Low computation complexity of online signature
- Per-message secret number
- ElGamal Signature Verification involves verifying that $g^x \bmod p = T_m T^{dm} \bmod p$, and it is faster than RSA based signature, making it important for smart card applications; the Digital Signature Standard is based on ElGamal.

A Zero-Knowledge Authentication

- Public key <n, v> with n being the product of two large primes
- Private s, with $v=s^2 \bmod n$, i.e., s is the square root of v

To prove to Bob that she is Alice:

1. Alice chooses k random numbers, $r_1, r_2, \ldots r_k$. For each r_i, she sends $r_i^2 \bmod n$ to Bob
2. Bob chooses a random subset of the r_i^2 and tells Alice which subset he has selected to be known as subset 1. The others will be known as subset 2.
3. Alice sends $sr_i \bmod n$ for each r_i^2 of subset 1, and sends $r_i \bmod n$ for each r_i^2 of subset 2.
4. Bob squares Alice's replies mod n. For those r_i^2 in subset 1 he checks that the square of the reply is $vr_i^2 \bmod n$. For those r_i^2 in subset 2 he checks that the square of the reply is $r_i^2 \bmod n$.

Why does this work?

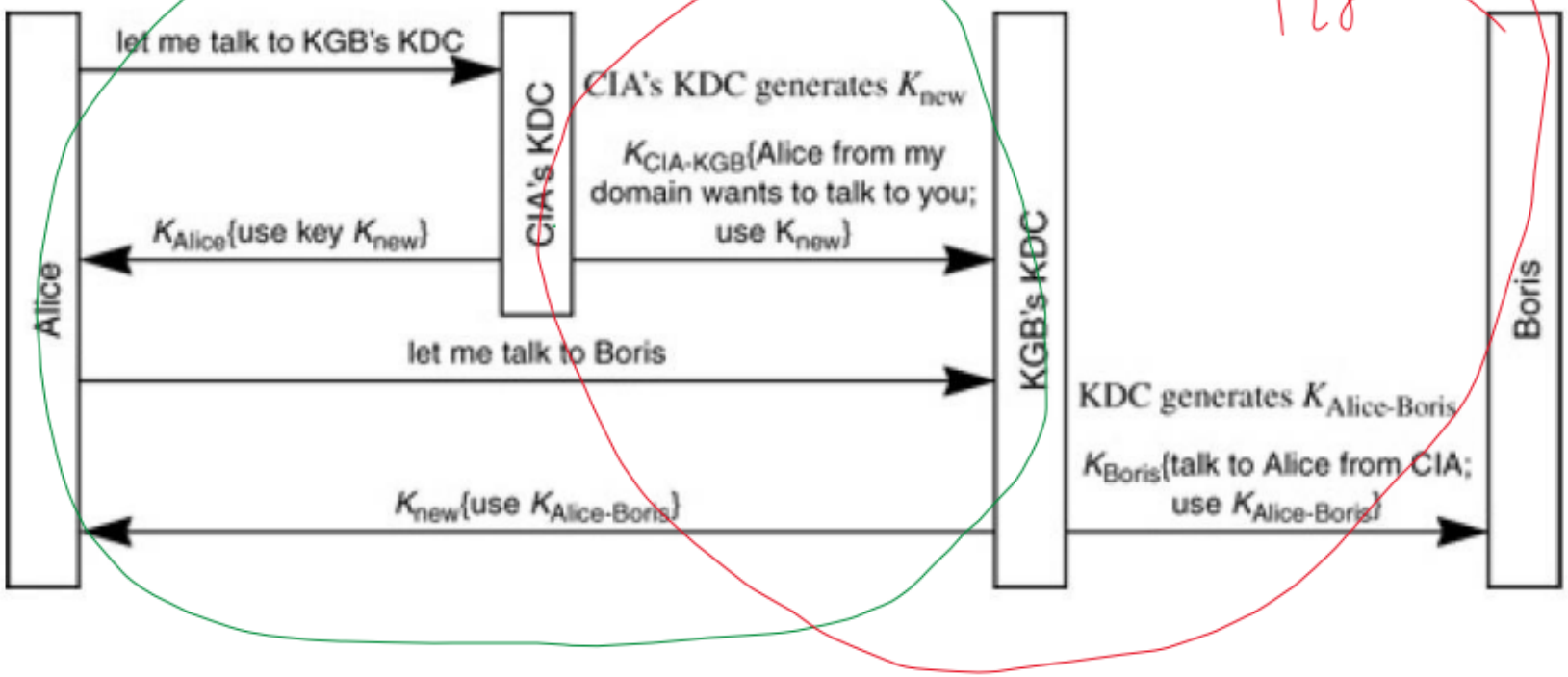
Finding square roots mod n is at least as hard as factoring n. This means that if you knew an easy way to find square roots mod n, you'd be able to factor n.

The voting process involves authentication to a verifier, creating a key pair and certification, and posting the vote using that key pair and certification.

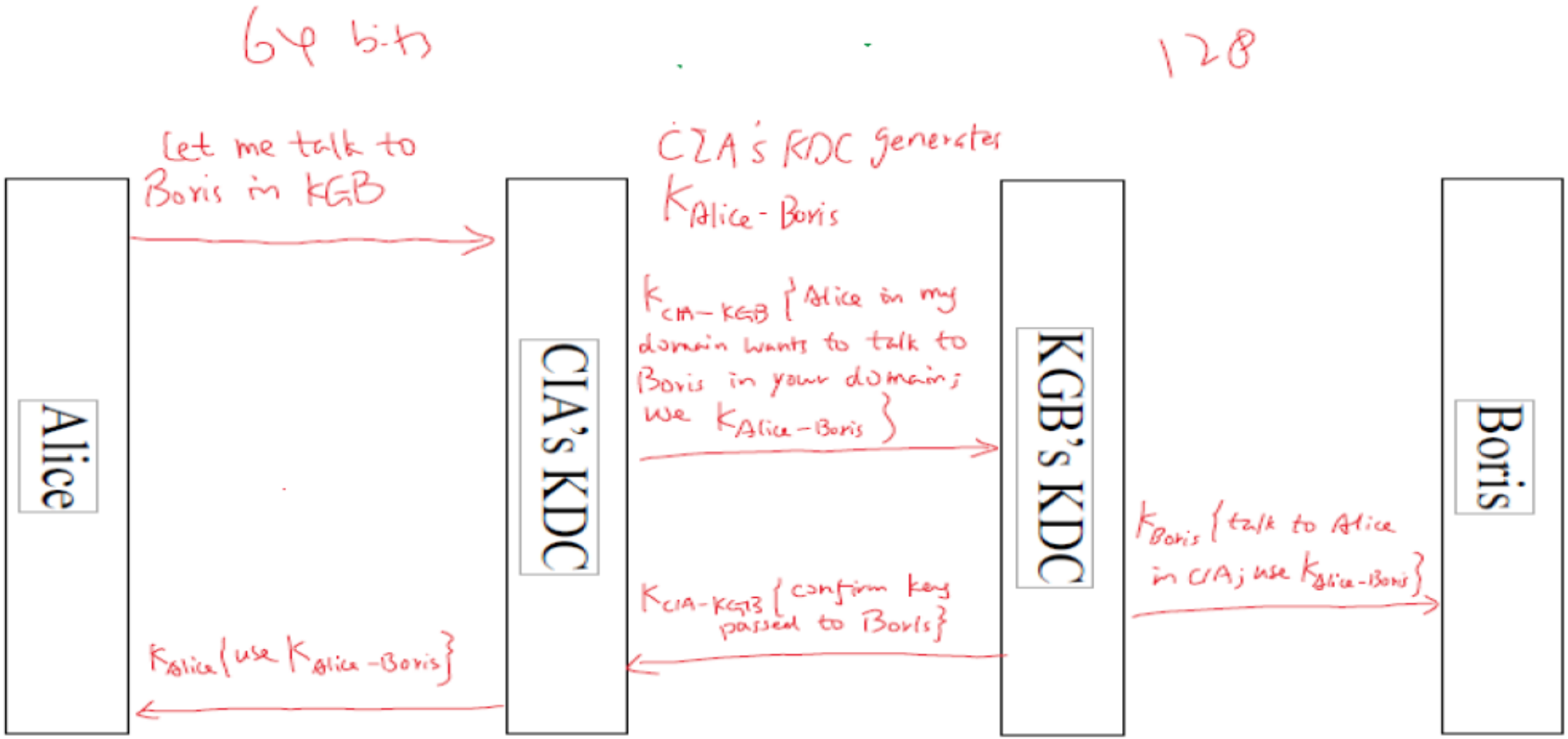
Blind Signatures

- Reg (voting registrar) signs certificates for (public, private) key pair anonymous certificate
- But Reg shouldn't know who goes with what key
- How can Reg sign, using RSA, without knowing what he's signing?
  - Reg's public key is (e,n).
  - Choose random R, raise to e.
  - Multiply message to be signed by R^e.
  - Have Reg sign it, which means you'll get R * signed msg.

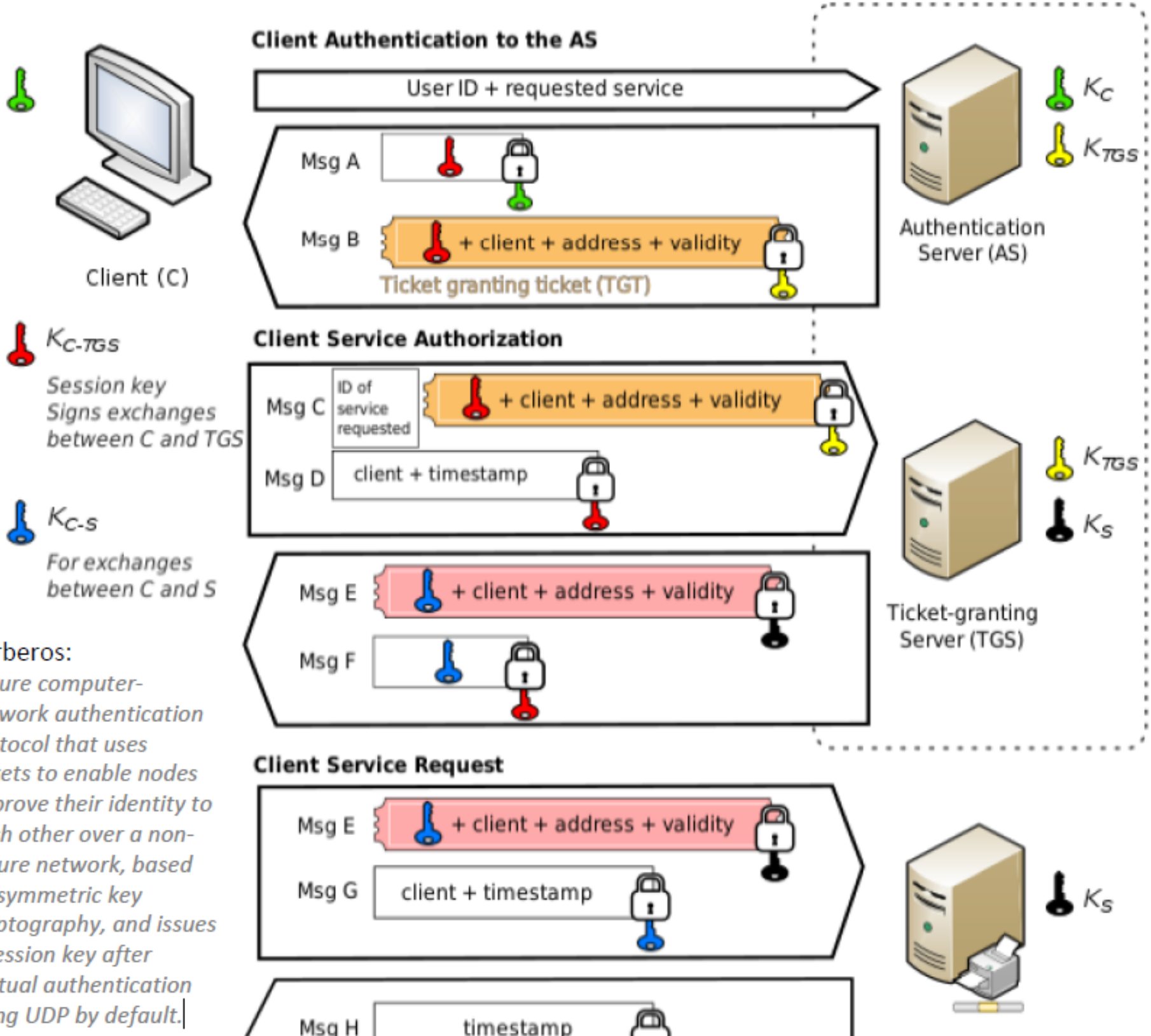
KDC Chain: Iterated Mode



KDC Chain: Recursive Mode



Recursive mode in Key Distribution Centers (KDCs) poses several challenges such as difficulty in maintaining policies for downstream domains, uncertainty in ensuring actions in further hops, and scalability issues due to the need to maintain status for each ongoing key establishment request.



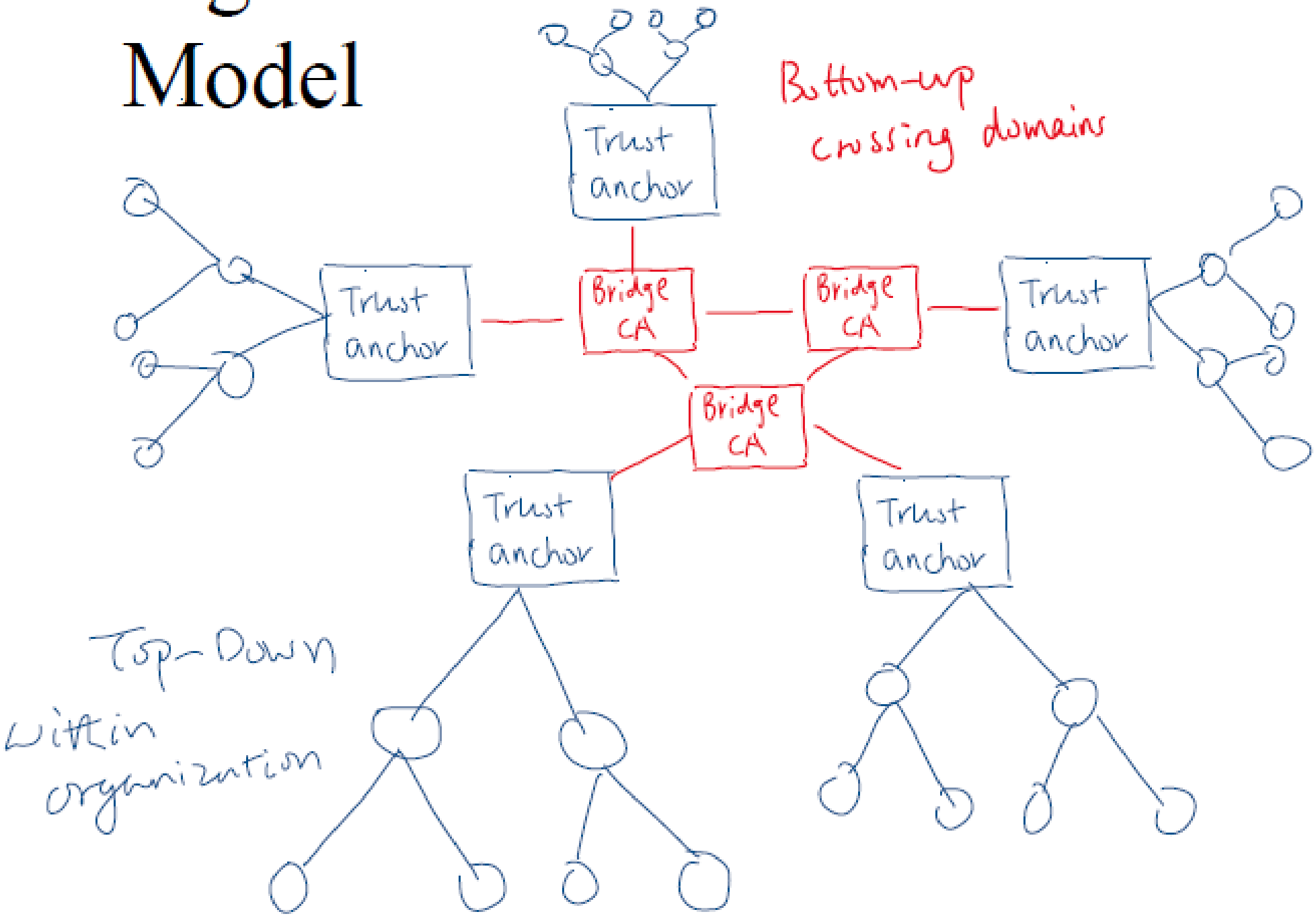
Kerberos:

Secure computer-network authentication protocol that uses tickets to enable nodes to prove their identity to each other over a non-secure network, based on symmetric key cryptography, and issues a session key after mutual authentication using UDP by default.

KDC vs CA Tradeoffs

- Impact of theft of KDC database vs CA private key
  - KDC: impact both before and after the theft
  - CA: impact only after the theft
- What needs to be done if CA / KDC compromised?
  - KDC: reestablish all keys and protecting all previous messages
  - CA: trace and handle invalid certificates since compromised
- Protecting KDC vs Protecting CA
  - KDC has to be on-line; CA does not need to be always on-line
  - **CA more likely behind firewall; KDC (e.g., Kerberos) uses UDP by** default (TCP optional), blocked many firewalls
- What if KDC vs CA down temporarily?
  - CA being down: new certs can't be issued; existing certs work fine.
  - KDC down: nobody can get a new session key for communications

Bridge CA Model



The Bridge CA model is a hierarchical system within an organization, but it also allows for cross-domain connections. The trust anchor is the root CA for the organization, which then points to the bridge CA, which in turn points to other organizations' roots.

Directories and PKI

- PKIX and X.509
  - X.509 specifies, amongst other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.
  - PKIX is a profile of X.509, which specifies which X.509 options should be supported.
- Directory
  - Name hierarchy
  - Store certificate
- Get public key from the directory instead of communicating with the other end first
- There are alternative certificate standards to X.509, including PKIX which avoids making hard decisions and allows for anything, and SPKI which rejects X.509 syntax and was merged with SDSI, which emphasizes linked local name spaces over a hierarchical global name space.

Revocation Problem

- The revocation problem refers to the challenge of revoking access when a password or smart card is stolen. With KDC, leaking a one-time session key will not cause future damage, but the long-term key must be reestablished.
- Tickets can have short lifetimes;
- Certificates have expiration dates, but it is inconvenient to renew them frequently
  - If sufficiently frequent and automated, CA can no longer be off-line
- Supplement certificate expirations with Certificate Revocation Lists (CRLs) or a blacklist server (OLRS)
- An online revocation server is less risky than an online certificate authority because it can only fail to report a revoked certificate, making the damage more contained.

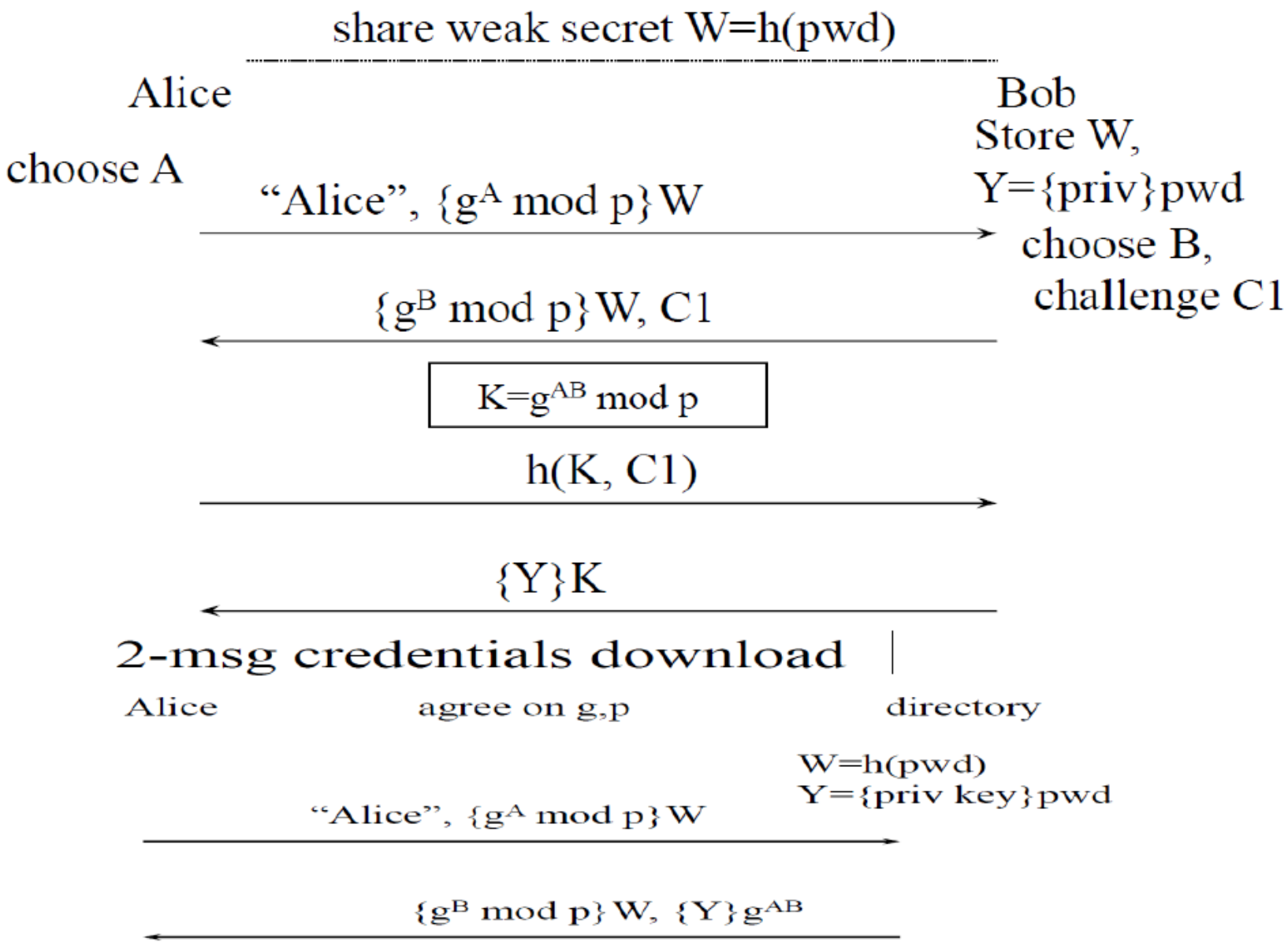
Interesting revocation ideas

- Incremental certs; full list also updated periodically
  – CRL not as secure as CA
  – Expired certificates can be removed from CRL
- Micali's hashing scheme
  – "Good" label with a refreshing schedule by hash
  – Mitigating overhead to check good guys
- "first valid cert", proposed by the textbook authors
  – No expiration date
  – Massive revocation
- good lists vs bad lists
  – Good list: scalability issue, privacy issue
  – Bad list: collusion with bribed CA

Obtaining Human's Private Key

- Encrypt the private key using the password as a key
  - Put the encrypted private key on a disk or a smart card
    - Portability, storing the disk, convenience to retrieve a key
    - Smart card may have limited resource to offer strong protection
  - Post the encrypted private key in a public place
  - Post the encrypted private key on a server that limits pwd guessing with a careful credential downloading protocol
  - The paper proposes a simplified method for downloading credentials by storing the private key and password on the server and sending them in two messages, without the need for a proper downloading protocol or authentication. The method can also include features like salt and saving server computation.

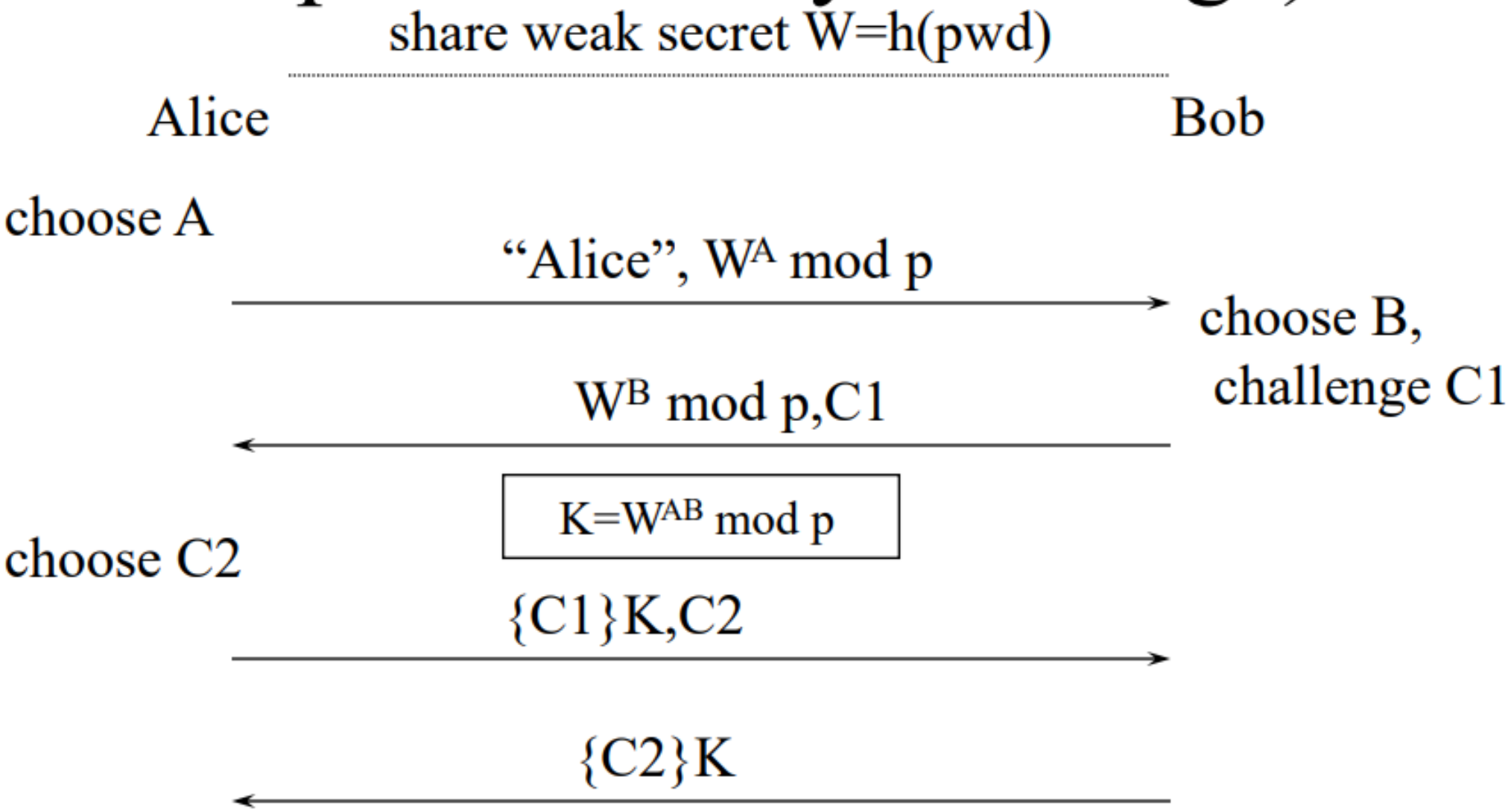
4-msg credentials download



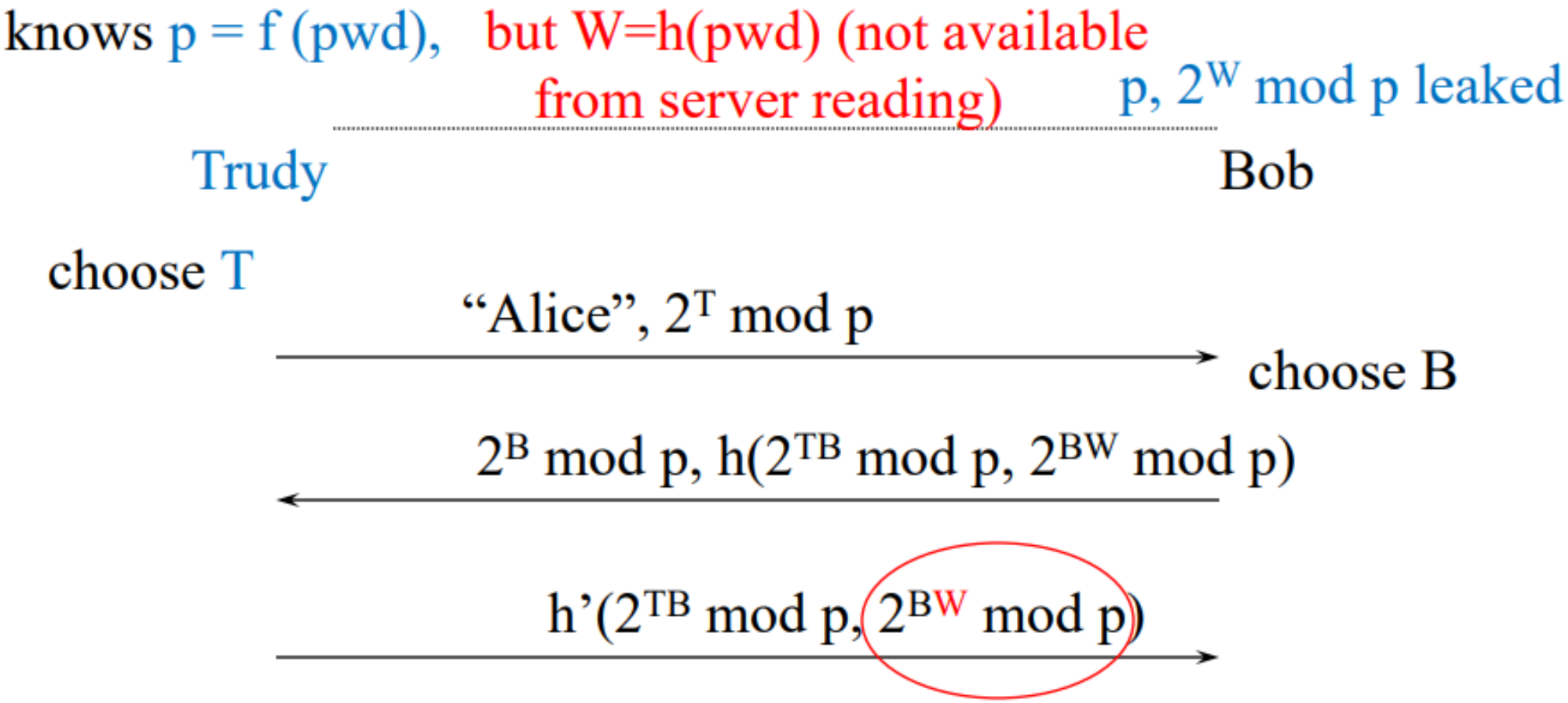
So what can we do about server database theft?

- "augmented" EKE (and SPEKE and PDM)
- Augmented EKE came up with the idea, but the protocol is complicated
- We'll show something simpler that has the same properties (and fewer msgs)
- Main idea: store something at server that can authenticate pwd
- Don't ever give server a pwd-equivalent
  - **When server reading happens, protection is at the level of dictionary based analysis**

SPEKE (simple password exponential key exchange)

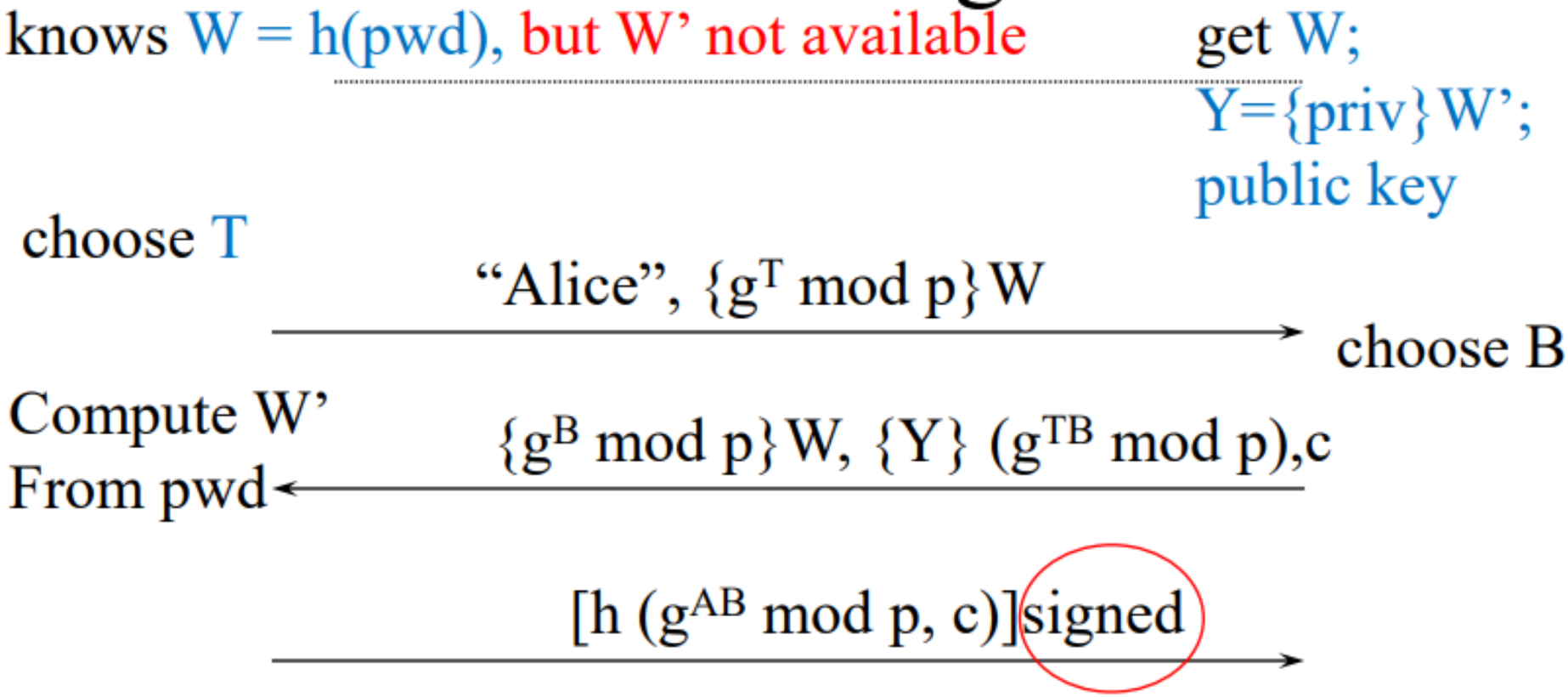


Augmented PDM: server reading



- Trudy cannot accomplish this step without knowing W
- Needs dictionary attack to hack password

RSA-Augmented EKE: server reading



- Trudy can decode Y, but cannot get "priv" without knowing W'
- Needs dictionary attack to hack password

SRP (Secure Remote Pwd)

compute W=h(pwd) — stores $g^W \bmod p$

Alice — Bob

choose A

"Alice", $g^A \bmod p = Y$ →

- can locally compute $g^W \bmod p$

← $Z = g^B + g^W \bmod p$, u, c1

choose B.
challenge c1,
32-bit # u

$K = g^{B(A+uW)} \bmod p$

- Receiving Z

$g^B \bmod p = (Z - g^W) \bmod p$

$K = (g^B \bmod p)^A \cdot (g^B \bmod p)^{uW} \bmod p$

Authentication requires explicit knowledge of W

Receiving Y

$K = (Y)^B \cdot (g^W \bmod p)^{Bu} \bmod p$

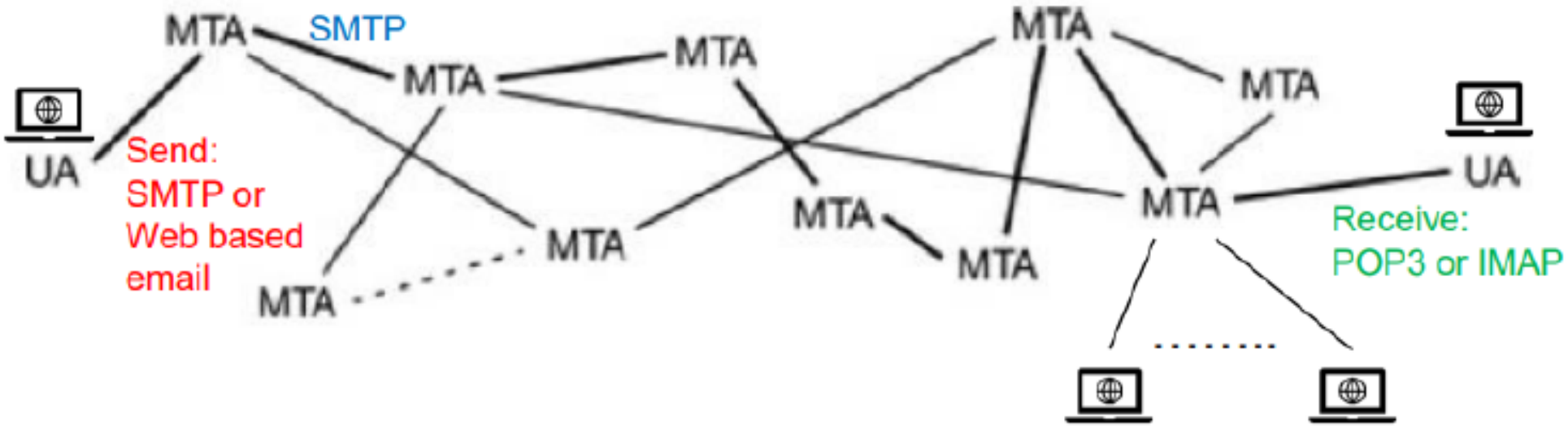
$= g^{AB} \cdot g^{WBu} \bmod p$

$= g^{B(A+uW)} \bmod p$

Complexities of email

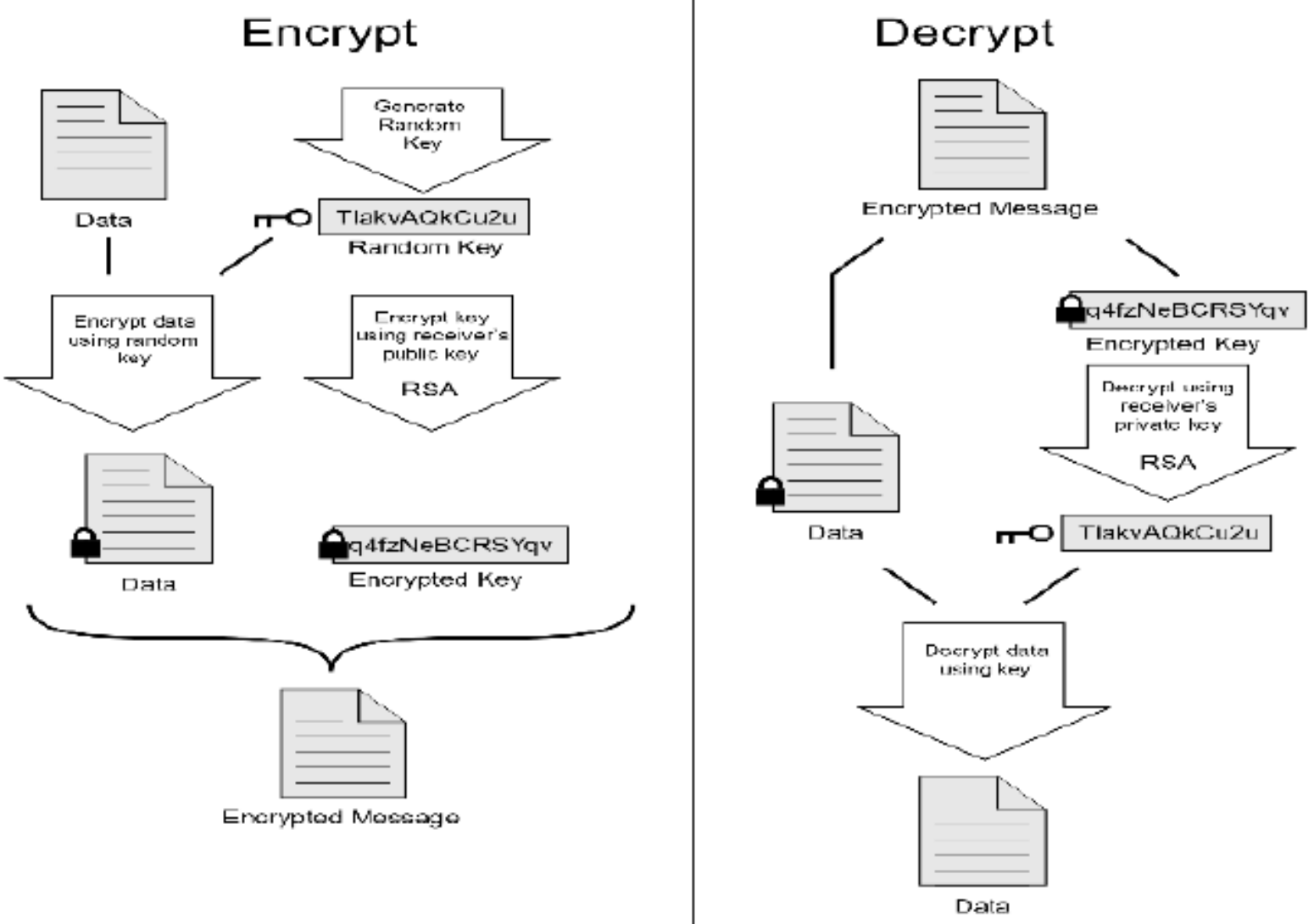
- text-based: no uniformed canonical text format
- distribution lists: one-to-many communications
  - The local exploder has advantages such as preventing mail forwarding loops, avoiding duplicate copies to individuals on multiple lists, and enabling easier estimation of bandwidth usage.
  - The advantages of a remote exploder include the ability to send to a list without knowing the membership, the efficiency of sending one copy to a distribution list maintainer, and the ability to parallel exploit a tree of distribution lists, which is particularly useful for large lists.
- store-and-forward: security crossing multiple hops
  - Heterogeneous systems along the path
  - Avoid unnecessary processing for integrity
  - Necessary forwarding processing
- The process of establishing keys can involve obtaining Bob's public key through a secure out-of-band mechanism or PKI, and Alice sending her certificate to Bob with a signed message requesting his public key. Secret keys can also be established through the use of KDC.

Mail Infrastructure:Store and Forward



- Allow intermittent connection
- A chain of Mail Transfer Agents (MTAs) to destinations
- Different MTA may use different protocol suites
- Share workload

Pretty Good Privacy (PGP)



Privacy involves protecting communication from eavesdropping in the network or relay node, considering design limitations of public key encryption for long messages to multiple recipients. Secret key encryption is preferred, but long-term keys are not desirable due to key "wear out" and potential playback attacks. It's important to protect conversations even if a long-term key is compromised and to consider involving an untrusted entity for one session.

Authentication

- Challenge with multiple recipients
- Public key, straightforward
- Secret key: various possibilities
  - CBC residue computed with per-user shared key.
  - keyed hash with per-user shared secret
  - MD encrypted with per-user shared secret
- Which secret key method to choose if there are multiple recipients?
  - To authenticate email with multiple recipients, use a symmetric key encryption method to encrypt the message with a shared key and then use public-key cryptography to securely transmit the key to each recipient individually.

Authentication with Distribution List

- Public key, straightforward
- Secret key, more complicated
  - Distribution list exploder authenticates Alice, Distribution list exploder use its own authentication information with each recipient, and not strong safe model.
  - The recipient will have to take the exploder's word that the message did originate from Alice
- Other interesting issues: non-repudiation vs plausible deniability
  - Public key based approaches
  - Secret key based approaches
- proof of submission / delivery
  - Over Internet, message deliver and receiver will not physically see each other (not the case in normal mail)
- Message flow confidentiality, Anonymity, Verifying the sending time

Verifying the Sending Time

- Preventing backdating
  - A company allows electronic purchase orders, signed with private key; not responsible for orders after reporting key stolen
  - Somebody made some purchases, then report his/her key was stolen. How the company can prove the purchase orders?
    - Argument: Bad guy stealing the key made the order but backdate to an early date
  - Have a notary date and sign the message, certificate, and CRL
- Preventing postdating
  - Postdating: a previous message attached with a current date
    - Postdate a previous 70% off coupon to a current date
  - Message needs to include some information only known at the sending moment
  - Message bonded with a signed time stamp from a notary

SSL vs TLS

- SSL and TLS are successive stages of one technology. The name was changed from SSL 3.1 to TLS 1.0 because of trademark issues with the old Netscape browser, for which SSL was originally developed.

- In late 2014, the PCI security standards organization reported, a vulnerability called POODLE (Padding Oracle On Downgraded Legacy Encryption) was rendering SSL (and even TLS 1.0) potentially ineffective

- The PCI DSS was done with SSL after June 30, 2016, for e-commerce applications.

Toy SSL: a simple secure channel

- *handshake:* Alice and Bob use their certificates, private keys to authenticate each other and exchange shared secret
- *key derivation:* Alice and Bob use shared secret to derive set of keys
- *data transfer:* data to be transferred is broken up into series of records
- *connection closure:* special messages to securely close connection

Security associations (SAs)

- before sending data, "security association (SA)" established from sending to receiving entity
  - SAs are simplex: for only one direction
- ending, receiving entitles maintain *state information* about SA
  - recall: TCP endpoints also maintain state info
  - IP is connectionless; IPsec is connection-oriented!
- how many SAs in VPN w/ headquarters, branch office, and n traveling salespeople?

IKE: PSK and PKI

- authentication (prove who you are) with either
  - pre-shared secret (PSK) or
  - with PKI (pubic/private keys and certificates).
- PSK: both sides start with secret
  - run IKE to authenticate each other and to generate IPsec SAs (one in each direction), including encryption, authentication keys
- PKI: both sides start with public/private key pair, certificate
  - run IKE to authenticate each other, obtain IPsec SAs (one in each direction).
  - similar with handshake in SSL.

Avoiding collisions (more)

idea: allow sender to "reserve" channel rather than random access of data frames: avoid collisions of long data frames

- sender first transmits *small* request-to-send (RTS) packets to BS using CSMA
  - RTSs may still collide with each other (but they're short)
- BS broadcasts clear-to-send CTS in response to RTS
- CTS heard by all nodes
  - sender transmits data frame
  - other stations defer transmissions

avoid data frame collisions completely using small reservation packets!

Selfish Misbehavior

- Normal behavior assumes that everybody follows the standard
- A malicious node may deliberately choose a smaller backoff timer and selfishly gain an unfair share of the network throughput
- Programmable wireless network device facilitate such an attack
- Unknown misbehavior strategy
- Real-time detection
- Without changing the standard

Detection with TCP/UDP Hybrid Traffic