

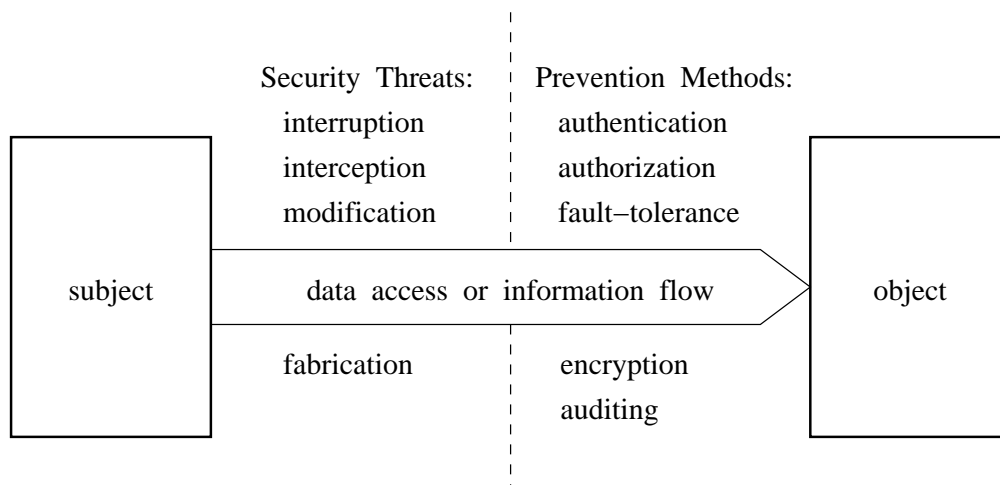
CHAPTER 8: DISTRIBUTED COMPUTER SECURITY

Security and dependability

- Confidentiality - protection from unauthorized disclosure of objects or activities
- Integrity - protection from unauthorized modification of data
- Availability - protection from denial of service (DoS)
- Reliability - tolerance of system faults
- Safety - tolerance of user faults

Security attacks

- External/internal intrusions
- Interruption, interception, modification, fabrication



AAA lines of defense

1. Authentication - outside intruders
2. Authorization - inside intruders
3. Auditing - passive deterrence

Security policy, model, and mechanism

- Policy - user requirements
- Model - formal representation of policies
- Mechanism - protection enforcement

- Separation of policy and mechanism
- Security servers and trusted kernel

Authentication - proof of identity

What you...

- Have (token, key)
- Are (biometrics)
- Know (password, crypto key)

Variants

- Weak authentication
- Strong authentication (ZKP)

Authorization - access control

- mandatory - systemwide, e.g., multilevel security
- discretionary - individual, e.g., access control matrix

Discretionary access control models

Access Control Matrix (ACM)

- Subjects: identity, class, or role-based
- Objects: subjects can also be objects
- Rights: access, transfer or copy

- ACM a sparse matrix
- Reducing the size by using groups and categories
- Distributed compartments
- Reference monitor

Access control matrix examples

subj. \ obj.	file a	file b	file c	file d
user A	owner	read / write	execute	owner
user B	copy read	owner		
user C		read	owner	append

(a) Resource ACM

subj. \ obj.	process A	process B	process C
process A		send / unblock	send / unblock
process B	receive		block
process C	receive	block	

(b) Process ACM

subj. \ obj.	domain A	domain B	domain C
domain A		enter	
domain B			enter
domain C	enter		

(c) Domain ACM

Implementation of ACM

- Access Control List (ACL) - by column per object, reservation system
- Capability List (CL) - by row per subject, ticket system
- Lock-Key - combination of ACL and CL, the Amoeba example

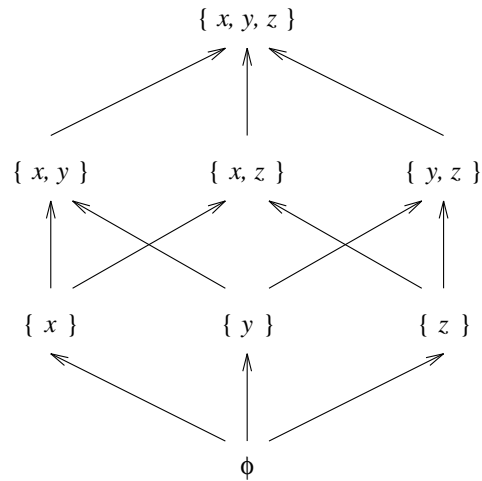
Comparison of ACL and CL

- Authentication
- Review of access rights
- Propagation of access rights
- Revocation of access rights
- Conversion between ACL and CL

Mandatory access control models

Lattice Model

- information flow model
- Direct Acyclic Graph (DAG)
- single source and single sink
- unique least upper bound and greatest lower bound



- Reflexive
- Transitive
- Antisymmetric

Multilevel security

- security class for subjects and objects
- security clearance and classification
- hierarchical security level and nonhierarchical security category

Bell-LaPadula model

- confidentiality
- no read up
- no write down

Biba model

- integrity
- no write up
- no read down

Cryptography

Goals:

- confidentiality of messages
- integrity of messages
- authenticity of subjects and objects
- anonymity

Cryptosystems

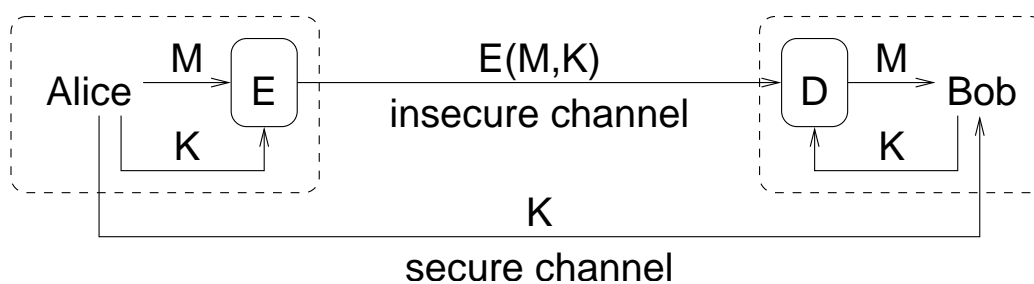
Issues:

- Codes vs. ciphers vs. steganography
- Block ciphers vs. stream ciphers
- Symmetric vs. asymmetric ciphers
- Kirchhoff's principle
- Key management

Symmetric (secret-key) systems

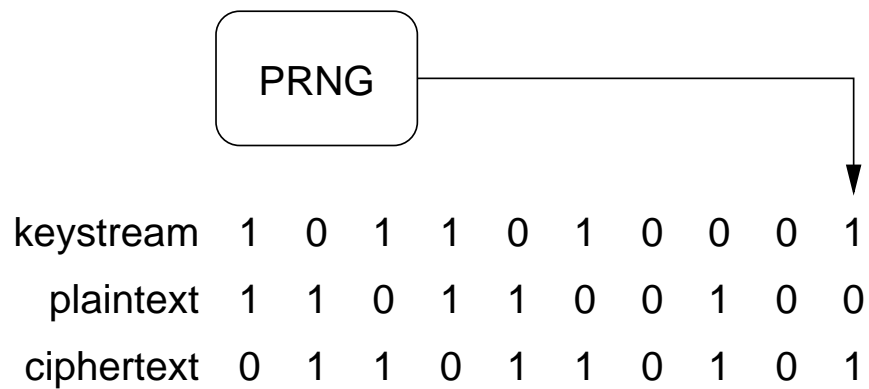
- single secret key, known algorithm
- symmetric, $M = D_{K_s}(E_{K_s}(M))$
- computational complexity
- number of keys, nature of key space
- stream ciphers - One-Time Pad (OTP), Vernam cipher
- substitution, transposition, concatenation of these
- modern block ciphers based on multiple rounds of concatenation
 - Feistel Structure, e.g., DES
 - Substitution-Permutation network, e.g., AES
- block cipher modes (ECB, CBC, OFB, CFB, CTR)
- key distribution

Symmetric Cryptography System Model



Stream Ciphers

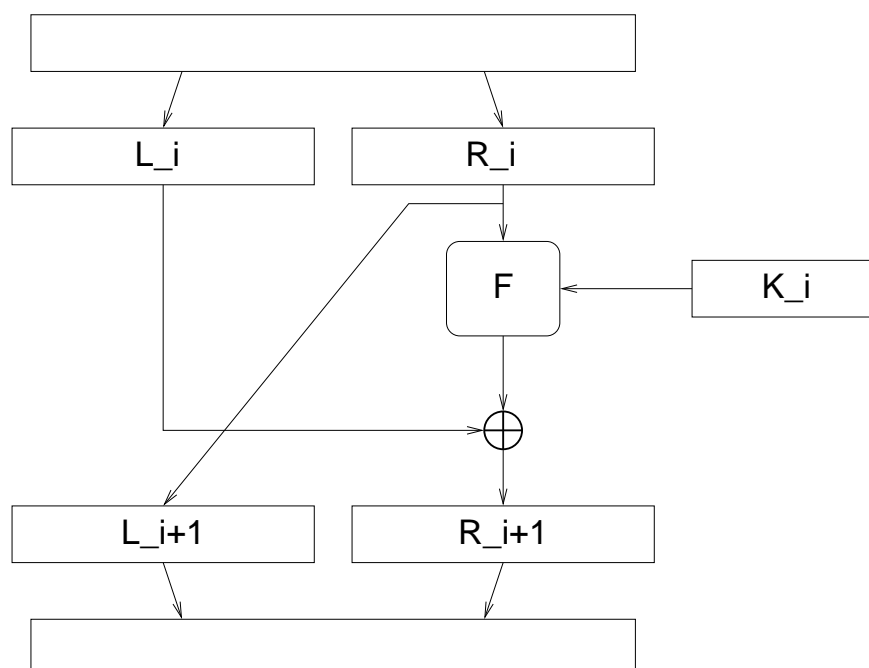
- low latency: symbol in (p_i), symbol out (c_i)
- $c_i = E(k_i, p_i)$ using key stream $k_1, k_2, \dots, k_i, \dots$
- OTP - $c_i = k_i \oplus p_i$ at bit level
- OTP - provably secure, but where to get random stream?
- OTP - key stream distribution
- Vernam cipher - use pseudorandom number stream
- block cipher stream modes (OFB, CFP, CTR)



Feistel Structure

- Block cipher that proceeds in N rounds
- Key K generates round keys K_i
- Break input plaintext block P into left L_0 and right R_0 halves
- In each round, scramble R_i using K_i , XOR with L_i to get R_{i+1} , L_{i+1} is R_i
- Output C is L_N concatenated with R_N

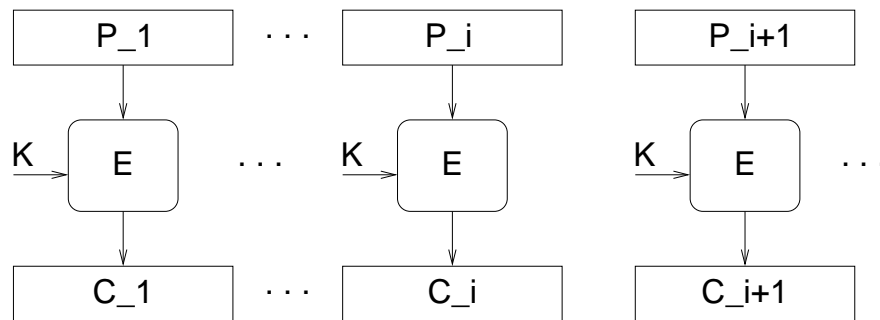
Feistel Structure



Block Cipher Modes

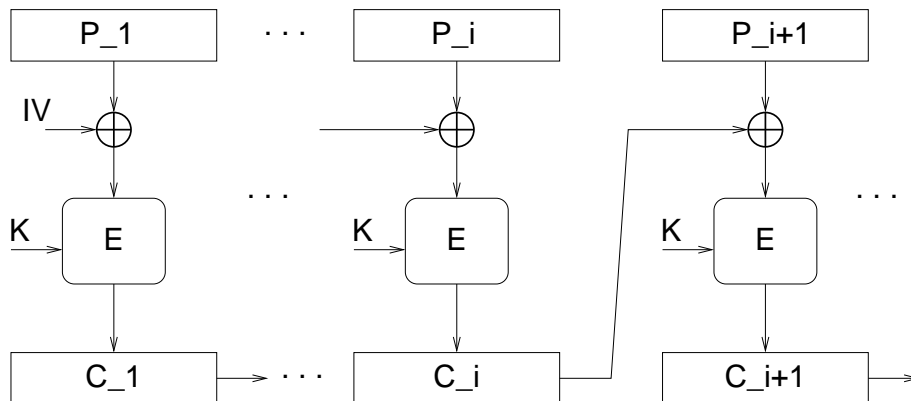
- ECB - Electronic Code Book: $C_i = E(K, P_i)$

Electronic Code Book (ECB) Mode



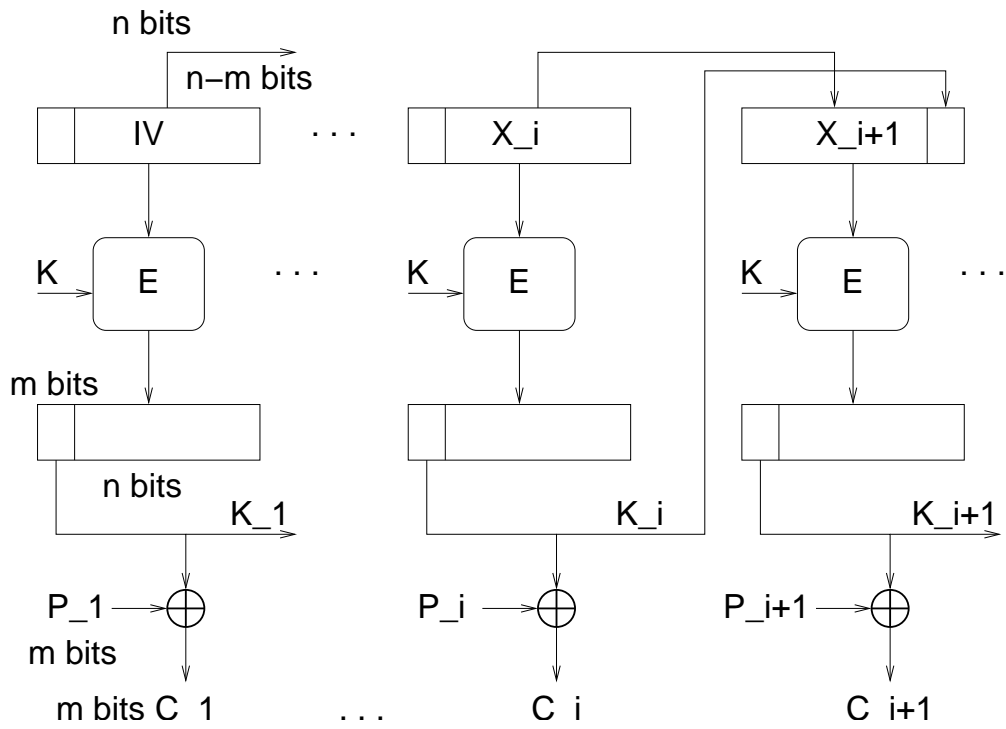
- CBC - Cipher Block Chaining: $C_i = E(K, P_i \oplus C_{i-1})$, $C_0 = IV$ (initialization vector)

Cipher Block Chaining (CBC) Mode



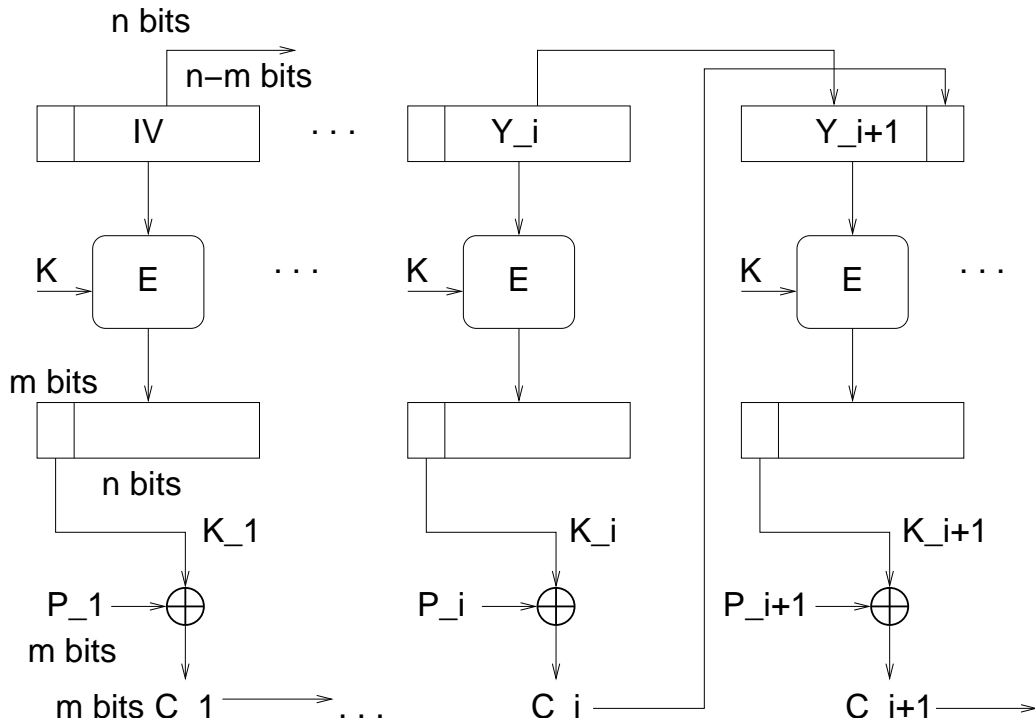
- OFB - Output Feedback (stream): $k_i = MSB_m(E(K, X_i))$
 $X_i = LSB_{n-m}(X_{i-1}) \ll m + k_{i-1}$, with $X_1 = IV$

Output Feedback (OFB) Mode

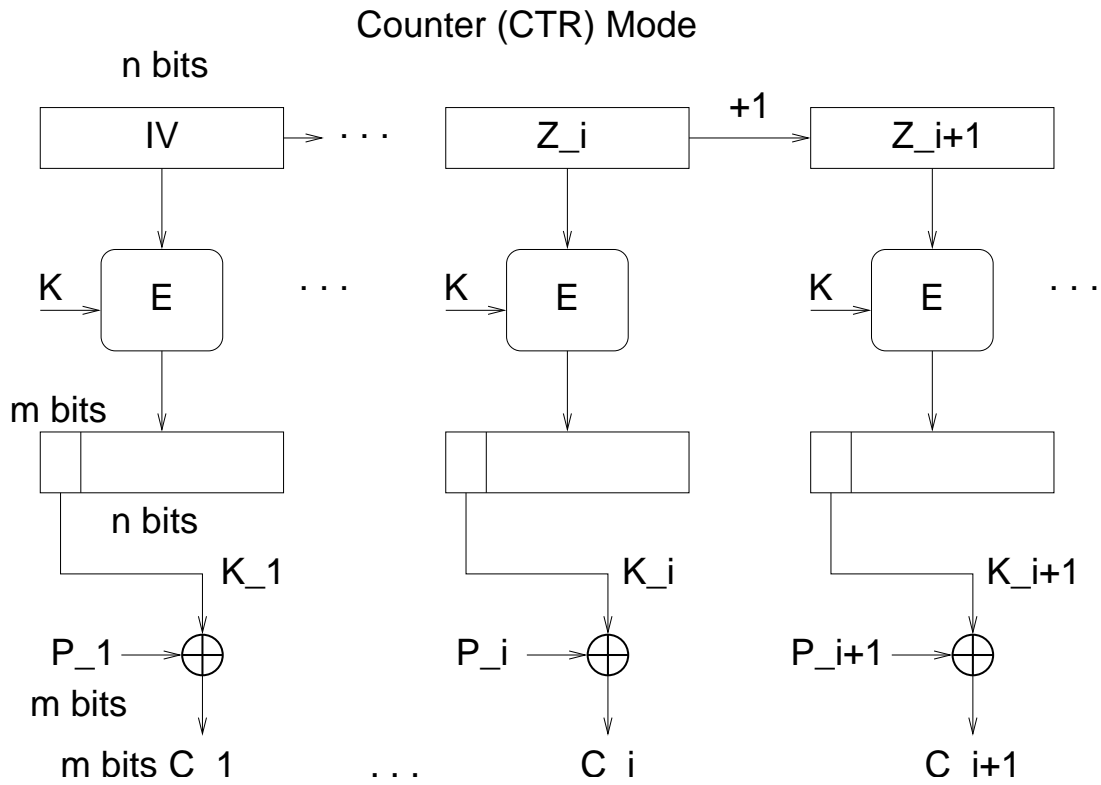


- CFB - Cipher Feedback (stream): $k_i = MSB_m(E(K, Y_i))$
 $Y_i = LSB_{n-m}(Y_{i-1}) \ll m + c_{i-1}$, with $X_1 = IV$

Cipher Feedback (CFB) Mode



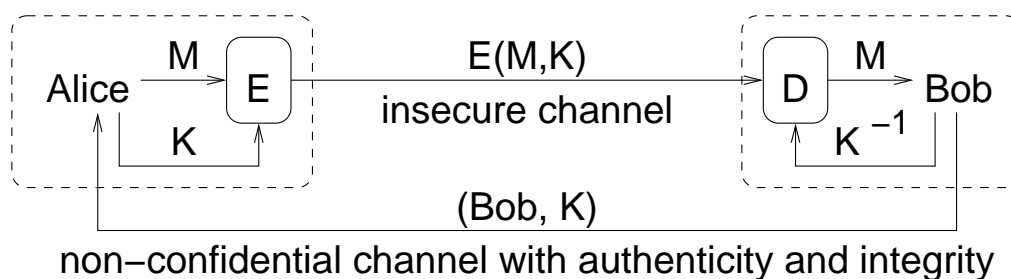
- CTR - Counter ((stream): $k_i = MSB_m(E(K, Z_i))$
 $Z_i = Z_{i-1} + 1$, and $Z_1 = IV$



Asymmetric (public-key) systems

- dual (public and secret) keys, known algorithm
- asymmetric, $M = D_{K_s}(E_{K_p}(M)) = D_{K_p}(E_{K_s}(M))$
- prime factoring, RSA
- discrete log, Diffie-Hellman
- discrete elliptic curve (ECC)
- computational complexity
- simpler key management
- certification of public keys

Asymmetric Cryptography System Model



Cryptographic Hashes

- Hash function (variable length \rightarrow fixed length)
- One-way (difficult to invert)
- Desirable for integrity
- Desirable for signing using public key cryptosystems
- Examples: MD5, SHA-1, SHA-256

Putting it all together: A PEM example

$E_{BIK_P}(DEK)$	$E_{DEK}(M)$	$E_{AIK_S}(MIC)$
RSA	DES	MD5

- Hash message M to produce MIC
- Encrypt MIC with A 's secret signing key, AIK_S to form signature
- Encrypt M with random, fresh, data encryption key, DEK
- Encrypt DEK with B 's public encryption key, BIK_P
- May include multiple recipients by encrypting DEK for each one
- RIPEM, CMS standards

Authentication and key distribution

Mutual Authentication

- A believes in B, B believes in A
- A believes that B believes in A
- B believes that A believes in B
- trusted authentication server

Requires:

- mutual secret or
- authentic public key info
- freshness
- demonstration of knowledge

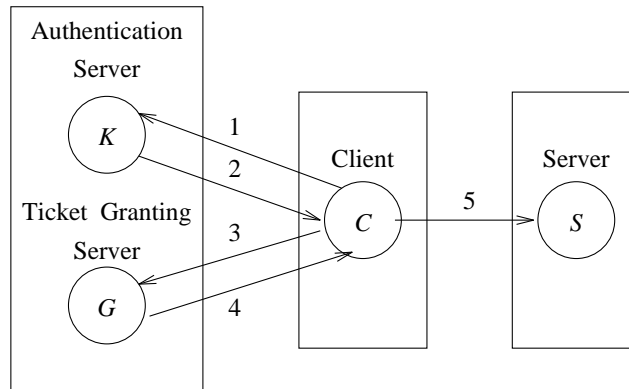
Needham-Schroeder authentication protocol

1. $A \rightarrow S$: A, B, N_a
2. $S \rightarrow A$: $\{N_a, B, K_{ab}, \{A, K_{ab}\}_{K_{bs}}\}_{K_{as}}$
3. $A \rightarrow B$: $\{A, K_{ab}\}_{K_{bs}}$
4. $B \rightarrow A$: $\{N_b\}_{K_{ab}}$
5. $A \rightarrow B$: $\{N_b - 1\}_{K_{ab}}$

- replay attacks
- freshness of messages
- nonce: number used only once
- challenge and response
- clock-synchronized, protected

Kerberos protocol

1. $A \rightarrow S$: A, B
2. $S \rightarrow A$: $\{K_{ab}, Ticket_{ab}\}_{K_{as}}$, where $Ticket_{ab} = \{B, A, addr, T_s, L, K_{ab}\}_{K_{bs}}$
3. $A \rightarrow B$: $Authenticator_{ab}, Ticket_{ab}$, where $Authenticator_{ab} = \{A, addr, T_a\}_{K_{ab}}$
4. $B \rightarrow A$: $\{T_a + 1\}_{K_{ab}}$



1. $C \rightarrow K$: C, G, N
2. $K \rightarrow C$: $\{K_{cg}, N\}_{K_c}, Ticket_{cg}$
3. $C \rightarrow G$: $Authenticator_{cg}, Ticket_{cg}$
4. $G \rightarrow C$: $\{K_{cs}, N\}_{K_{cg}}, Ticket_{cs}$
5. $C \rightarrow S$: $Authenticator_{cs}, Ticket_{cs}$

Proxy and delegation

1. $A \rightarrow B$: $[R, PS_{proxy}]_A, K_{A,B}(SS_{proxy})$
2. $B \rightarrow S$: $[R, PS_{proxy}]_A$
3. $S \rightarrow B$: $PS_{proxy}(N)$
4. $B \rightarrow S$: N

Some advanced topics

- repeated authentication
- complex security policies
- proxy protocols
- covert channel
- monitoring and auditing