- **Buffer Overflow Attack** is kind of Denial of Service. Inject attack code into buffer, redirect control flow to attack code and execute it.
- **Denial of Service** make a network service unusable, usually by overloading the server or network.

  Consume host resources using TCP SYN floods or ICMP ECHO (ping) floods, or consume bandwidth using UDP floods or ICMP floods.
- **Computer Viruses** is parasitic programs which are designed to alter the way a computer operates without the permission or knowledge of the user.

  It executes and replicate itself. It will often place its own code in the path of execution of another program.

  They incorporate themselves within executable program files or infects in files. Some attach themselves to boot records.
- **Computer Worm** create back doors into your systems, allowing unauthorized access and consume local system resources.
- **Computer Virus vs Worm**

  Viruses are designed to spread themselves from a file to another on a computer. It depends on human aids, that is, it required infected host file, but worms don't.

  Worms are designed to spread themselves form one computer to another over a network. It doesn't need help from human being.
- **Port Scan** is reconnaissance techniques used to discover vulnerable machines or services they can break into.

  Responses are different depending on whether the selected service is running or not.
- **TCP SYN Scan**

```
                                    [OPEN]
SRC  SYN + Port 80------------->    TRG
     <------------------SYN/ACK
     RST----------------------->

                                    [CLOSE]
SRC  SYN + Port 113------------>    TRG
     <----------------------RST
```

- **TCP FIN Scan**

```
                                    [OPEN]
SRC  FIN + Port 23------------->    TRG
     x-------------[NO RESPONSE]

                                    [CLOSE]
SRC  FIN + Port 443------------>    TRG
     <----------------------RST
```

- **TCP Null Scan**

```
                                    [OPEN]
SRC  0x00 + Port 110----------->    TRG
     x-------------[NO RESPONSE]

                                    [CLOSE]
SRC  0x00 + Port 448----------->    TRG
     <----------------------RST
```

- **TCP Xmas Tree Scan**

```
                                    [OPEN]
SRC  FIN, URG, PUSH + Port 79-->    TRG
     x-------------[NO RESPONSE]

                                    [CLOSE]
SRC  FIN, URG, PUSH + Port 618->    TRG
     <----------------------RST
```

- **Pros & Cons of Above TCP Scan Approaches**

  There is no TCP session are completed for these scans, so, usually no logs on remote machines.

  There is minimal overhead: two packets for a closed port, one packet for an open port.

  Not applicable to MS's implementation of the TCP/IP stack. On a Windows-based computer, all ports will appear to be closed regardless of their actual state.

  Raw socket capability is required to create these specialized packets. This requires privileged access to the system.

- **UDP Scan**

  There is no 3-way handshake in connectionless protocol. It accomplished by sending a single IDP packet to each port on the target system.

  Scanned system will return **ICMP "Port Unreachable"** packet if UDP packet is sent to a closed UDP port. It assumes an open port sends no ICMP packet.

  It is not a very accurate scan and very slow.

- **RPC Scan** looks for Remote Procedure Call services and their version numbers. Used in conjunction with other scan types.

- **Ping Scan** finds out which IPs in a given range of IPs or network are active. A.k.a. Ping Sweep.

- **Characteristics of Scanners**

  Usually they make a large number of connection attempts a second. But, it is not the case for **slow scanners** or **stealthy scanners**.

  Since they do not have a complete list of active IP addresses, they usually make more failed connection attempts than normal hosts.

- **Issues for Practical Scanning Detection System**

  Memory constraint: SRAM / Low false positive and low false negative rates / Low implementation complexity for packet processing at high link speeds / Fast detection / Detection of slow scanners.

- **Shobha's Scheme**

  k -superspreader: host that contacts at least k distinct destinations in short time period.

  Given stream of src-dst pairs, find k-superspreaders in the stream.: heavy distinct-hitter problem.

- **Jung's Scheme**

  정상적인 유저가 성공적인 시도를 할 확률이 높다.

  H_0: hypothesis that the given source r is benign
  H_1: hypothesis that the given source r is scanner

  Y_i: outcome of first connection attempt by r to i-th distinct host

  $Y_i = 0$ if the attempt is a success
  $Y_i = 1$ if the attempt is a failure

  $$\Pr(Y_i = 0|H_0) = \theta_0, \quad \Pr(Y_i = 1|H_0) = 1 - \theta_0,$$
  $$\Pr(Y_i = 0|H_1) = \theta_1, \quad \Pr(Y_i = 1|H_1) = 1 - \theta_1$$

  Assumption: $\theta_0 > \theta_1$.

  $$\Lambda(Y) = \frac{\Pr(Y|H_1)}{\Pr(Y|H_0)} = \prod_{i=1}^{n} \frac{\Pr(Y_i|H_1)}{\Pr(Y_i|H_0)}$$

  If $\Lambda(Y) \leq \eta_0$, then we select H_0.
  If $\Lambda(Y) \geq \eta_1$, then we select H_1.

  Example, $\theta_0 = 0.8$, $\theta_1 = 0.2$, $\eta_0 = \frac{1}{99}$, $\eta_1 = 99$

  If $Y = \{0, 1, 0, 1, 1, 1, 1, 1\}$

Y	0	1	0	1	1	1	1	1
Λ	1/4	1	1/4	1	4	16	64	256

  It is complex to implementation and memory size requirement are not analyzed.

  If the scanner knows some valid IP addresses, then it may evade detection. 일부러 알고있는 IP에 접근을 시도해서 Λ 값을 낮아지도록 함.

- **Detection Rule of Nam's Scheme: Sampling-Based**

  $$q(s) = \frac{\text{RESPONSE}(s)}{\text{ATTEMPT}(s)}, \qquad q(s) \leq \eta$$

d_1	d_2	$\cdots$	d_6	a_1	a_2	a_3	d_1	d_2	$\cdots$	d_6	a_5	a_6	a_7	$\cdots$
s	s		s		s	s						s		

  Randomly sample src-dst pair with a sampling probability p_s and consider the connection attempts of only sampled n src-dst pairs for detection.

- **Denial of Service (DoS)** is the act of performing an attack which prevents the system from providing services to legitimate users.

  When successful, the targeted host may stop providing any service, provide limited services only or provide services to some users only.

- **Smurf Attack** is a Distributed DoS (DDoS) in which large numbers of ICMP packets with the intended victim's spoofed source IP are broadcast to a computer network using an IP broadcast address.

  ICMP Echo Request 패킷의 Source 주소를 공격대상의 IP주소로 하고, Destination 주소를 브로드캐스트 주소로 하여 보내면, 근처의 컴퓨터가 ICMP Echo Reply 패킷을 공격대상의 IP주소로 보낸다.

- **Ping of Death**

  If an ICMP packet with a payload > 64K is received, machine will crash or reboot. Packets of this length are illegal, so programmers did not account for them.

- **Distributed Dos (DDoS) Attack**

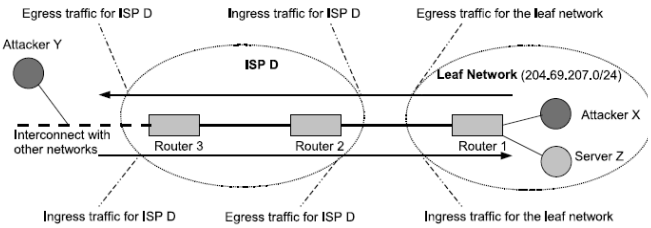
  With DDoS, every member of the attack generates relatively small amounts of traffic. The combined result overwhelms the remote system.

- **BotNets (Zombie Networks)** are networks of thousands of compromised hosts with dedicated software installed.

  A single controller can then be used to launch an attack from the entire network.

- **Storm Worm** is P2P-based botnet. It usually propagates by using e-mail.

- **Ingress Filtering**



  네트워크 내부의 공격자는 검출 불가능.

- **Hop-Count Filtering (HCF)**

  Since HCF tries to weed out spoofed IP packets, it helps to sustain the availability of the service during DDoS attack.

  It is very different hop-count between Attacking Machine's and Real Source Machine's

- **SYN Cookies**

  서버가 클라이언트로부터 SYN패킷을 받아도 서버에 자원을 할당하지 않고 SYN/ACK패킷에 Cookie를 함께 보낸다. (SEQ 필드에)

  서버는 ACK 패킷에 [Cookie+1]이 왔을 때, Cookie 값을 확인하고 클라이언트에 대한 자원을 할당한다.

- **CAPTCHA** is revers Turing test.

- **Computational Puzzles**

  Server S uses a secret key K to generate puzzle challenge and to verify authenticity of puzzle solution without keeping any state.

$$\text{Secret } T = \text{MAC}_K(\text{Client IP},\ \text{Client Port})$$

  Example: use last r bits of T as image.

$$S \rightarrow C: T, r$$

  C searches x s.t. last r bits match T

$$[H(x)]_r = [T]_r$$

  Then $C \rightarrow S: T, r, x$

- **Hash-Based IP Traceback**

  Routers keep information about which packets they forward, victims can query routers (whether they have forwarded packet) for traceback.

  Routers store ~512MB Bloom filter. For each packet, set a few bits in filter. For query, check if bits are set. No false negatives, but can make false positives.

  Single packet can be traceback, but large overhead, poor incremental deployment, possibility of new DoS attacks on the router by query messages.

- **Simple IP Traceback**

  Routers add their IP address to each forwarded packet, but it cannot pre-allocate space and increase length of packet causes more fragmentation and drop router performance.

- **Probabilistic Packet Marking**

  Mark packets in the Fragment Identification field with route information rather than generating extra packets. Victim reconstructs attack path using multiple of packets.

- **Pushback**

  Routers classify malicious aggregates and apply rate-limiting filters to them.

  Aggregates: Collection of packets with common properties such as destination, source, protocol, etc.

  Routers propagate filters upstream to limit bandwidth consumption by malicious aggregates.

- **Content Distribution Networks (CDNs)**

  Challenging to stream large files from single origin server in real time. CDN replicate content at hundreds of servers throughout Internet.

  Placing content close to user avoids impairments of sending content over long paths. CDN server typically in edge/access network.

  CDN simple content access scenario

  1. Bob gets URL for video URL L from webpage
  2. Resolve L via Bob's local DNS
  3. Provider's DNS returns URL L'
  4. Resolve L' via CDN's authoritative DNS, which returns IP address of CDN server with video.
  5. Request video from CDN server, then streamed.

- **CDN Cluster Selection Strategy**

  How does CDN DNS select good CDN node to stream to client.

  Geographically closest / Shortest delay / IP anycast

  Or, let client decide: give client a list of several CDN servers, client pings server, pick the bets.

- **Common Defense Mechanisms**

  Make a history information and monitoring the following 5 factors.

  Get & Post requests / Other request types / Outbound HTTP Mbps / Max number of request per source / Max number of request per connection

  If any real-time value of a factor is higher than the history's upper bound, it will be detected as an attack.

- **Anomaly Detection**

  Threshold-based technique that detects the clients who request same page more than a few times in a second.

  Model based on the "browsing order of pages" or "correlation with browsing time to page information size".

- **Examples of Intrusion**

  Remote root compromise / Web server defacement / Guessing or cracking passwords / Copying databases containing credit card numbers / Viewing sensitive data without authorization / Running a packet sniffer / Distributing pirated software / Using an unsecured modem to access internal network / Impersonating an executive to get information – calling the help d나 새 reset the executive's e-mail password and learn the new password / Using an unattended, logged-in workstation without permission

- **Intruder Behavior: Target Acquisition and Information Gathering**

  Explore corporate website for information on corporate structure, personal, key systems, as well as details of specific web server and OS used.

  Gather information on target network using DNS look up tools such as WHOIS DB

  Map network for accessible service using tools such as NMAP.

  Send query email to customer service contact, review response for information on mail client, server, and OS used, and also details of person responding. Identify potentially vulnerable services.

- **Intruder Behavior: Initial Access**

  Brute force a user's web content management system (CMS) password. Exploit vulnerable in web CMS plugin to gain system access.

- **Intruder Behavior: Privilege Escalation**

  Scan system for application with local exploit. Exploit any vulnerable application to gain elevated privileges.

  Install sniffers to capture administrator passwords. Use captured administrator password to access privileged information.

- **Intruder Behavior: Information Gathering or System Exploit**

  Scan files for desired information. Transfer large numbers of documents to external repository. Use guessed or captured passwords to access another server on network.

- **Intruder Behavior: Maintaining Access**

  Install remote administration tool or rootkit with backdoor for later access. Use administrator password to later access network. Modify or disable anti-virus or IDS programs running on system.

- **Intruder Behavior: Covering Tracks**

  Use rootkit to hide files installed on system. Edit logfiles to remove entries generated during the intrusion.

- **Security Intrusion** is a security event, or a combination of multiple security events, that constitutes a security incident in which an intruder gains, or attempts to gain, access to system (or system resource) without having authorization to do so.

  권한이 없는 자로부터 발생되는 시스템 또는 시스템의 자원에 접근 시도를 위한 보안 사고를 구성하는 사건 또는 사건의 조합.

- **Intrusion Detection** is a security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner.

- **Host-based Intrusion Detection System (HIDS)**

  Monitors the characteristics of a single host for suspicious activity.

  Adds a specialized layer of security software to vulnerable or sensitive systems. Can use either anomaly or signature and heuristic approaches.

  Primary purpose is to detect intrusions, log suspicious events, and send alerts. Can detect both external and internal intrusions.

  Common data sources include: System call traces / Audit (log file) records / File integrity checksums / Registry access

- **Network-based IDS (NIDS)**

  Monitors network traffic and analyze network, transport, and application protocols to identify suspicious activity.

  Monitors traffic at selected points on a network / Examines traffic packet by packet in real or close to real time / May examine network, transport, and application-level protocol activity

  Comprised of a number of sensors, one or more servers for NIDS management functions, and one or more management consoles for the human interface.

  Analysis traffic patterns may be done at the sensor, the management server or a combination of the two.

  Inline sensors are good for intrusion prevention. **Passive sensors** are no extra handling at router or switches (less on impact on packet delay).

- **Distributed or Hybrid IDS**

  Combines information from a number of sensors, often both host and network based, in a central analyzer that is able to better identify and respond to intrusion activity.

  May need to deal with different sensor data formats. One or more nodes in the network will serve as collection and analysis points for the data from the systems on the network. Thus, either raw sensor data must be transmitted across the network. Therefore, there is a requirement to assure the integrity and confidentiality.

  서로 다른 센서 데이터 형식을 처리할 수 있어야함. 한 개이상의 노드가 데이터 수집 및 분석 지점으로 사용됨. 따라서 센서 데이터를 네트워크상에서 전달해 줘야하므로 데이터 무결성 및 기밀성 보장 되어야함.

  중앙집중식 또는 분산형 구조를 사용할 수 있음.

- **DHIDS: Host Agent Module** is an audit collection module operating as a background process on a monitored system. Its purpose is to collect data on security-related events on the host and transmit these to the central manager.

- **DHIDS: LAN Monitor Agent Module** operates in the same fashion as a host agent module except that it analyzes LAN traffic and reports the result to the central manager.

- **DHIDS: Central Manger Module** receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion

- **Honeypots** are decoy systems designed to lure a potential attacker away from critical systems. Collect information about the attacker's activity.

  Encourage the attacker to stay on the system long enough for administrators to respond. Systems are filled with fabricated information that a legitimate user of the system wouldn't access.

  방화벽 앞에, 내부 네트워크 내부의 노드로, 서비스 네트워크 내부의 노드로 위치될 수 있음.

- **Low Interaction Honeypot** consists of a software package that emulates particular IT services or systems well enough to provide a realistic initial interaction, but does not execute a full version of those services or systems

  에뮬레이션 되어 초기 상호작용은 제공하지만, 정식 버전은 실행되지 않는 소프트웨어 패키지.

- **High Interaction Honeypot** is a real system, with a full OS, services and applications, which are instrumented and deployed where they can be accessed by attackers.

- **Snort** is a multi-mode packet analysis tool consist of Sniffer / Packet Logger / Forensic Data Analysis tool / Network Intrusion Detection System

  Packet Decoder / Preprocessor / Detection Engine / Logging and Alerting System

- **Firewall**

  All traffic from inside to outside, and vice versa, must pass through the firewall. Only authorized traffic as defined by the local security policy will be allowed to pass. The firewall itself is immune to penetration.

- **Firewall Capabilities**

  Defines a single choke point. Provides a location for monitoring security events. Convenient platform for several Internet functions that are not security related. Can serve as the platform for IPSec.

- **Firewall Limits**

  Cannot protect against attacks bypassing firewall. May not protect fully against internal threats. Improperly secured wireless LAN can be accessed from outside the organization. Laptop, PDA or portable storage device may be infected outside the corporate network then used internally.

- **Packet Filtering Discard Policy**

  Prohibit unless expressly permitted. It is more conservative, controlled, visible to users.

- **Packet Filtering Forward Policy**

  Permit unless expressly prohibited. Easier to manage and use but less secure.

- **Stateless Packet Filter**

  Admits packets that "make no sense" e.g. dest port = 80, ACK bit set, even though no TCP connection established.

  의미가 없는 패킷을 승인함.

- **Stateful Packet Filter**

  Track status of every TCP connection such as setup (SYN), teardown (FIN): determine whether incoming, outgoing packets "make sense".

  Timeout inactive connections at firewall: no longer admit packets.

- **Application-Level Gateway**

  Also, called an Application Proxy. User contacts gateway using a TCP/IP application, then, authenticate. Gateway contacts application on remote host and relays TCP segments between server and user. It tends to be more secure than packet filters.

  It needs additional processing overhead on each connection because each application must have proxy code and, needs proxy server for each application.

- **Circuit-Level Gateway**

  Sets up two TCP connections, one between itself and a TCP user on an inner host and one on an outside host.

  Relays TCP segments from one connection to the other without examining contents. Security function consists of determining which connections will be allowed.

  May use application-level gateway inbound and circuit-level gateway outbound. Circuit-level gateway has lower overheads.

- **Bastion Hosts** is the system identified as a critical strong point in the network's security.

  Runs secure OS, only essential services.
  May require user authentication to access proxy.
  Each proxy can restrict features, hosts accessed.
  Each proxy is mall, simple, checked for security.
  Each proxy is independent, non-privileged.
  Limited disk use, hence read-only code.

- **Host-Based Firewalls u**sed to secure an individual host. Common location is a server.

  Filtering rules can be tailored to the host environment. Protection is provided independent of topology. Provides an additional layer of protection.

- **Personal Firewall** typically much less complex than server-based or stand-alone firewalls. Primary role is denying unauthorized remote access.
- **Internal Demilitarized Zone (DMZ) Network**

  외부에서 접근 할 수 있지만 보호가 필요한 시스템은 대개 이 네트워크에 있다. 일반적으로 회사 웹 사이트, 메일 서버 또는 DNS 서버와 같은 외부 연결이 필요한 서버가 여기에 있다.
- **Host-resident Firewall** includes personal firewall software and firewall software on servers.
- **Screening Router** is single router between internal and external networks with stateless of full packet filtering
- **Single Bastion Inline** is single firewall device between an internal and external router.
- **Single Bastion T** has a third network interface on inline bastion to DMZ where externally visible servers are placed.
- **Double Bastion Inline**: DMZ is sandwiched between bastion firewalls.
- **Double Bastion T**: DMZ is on a separate network interface on the bastion firewall.
- **Distributed Firewall Configuration** used by large businesses and government organizations.
- **Symmetric Key Distribution Ways**

  "A" can select a key and physically deliver it to "B".

  "A" third party can select the key and physically deliver it to A and B

  If "A" and "B" have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.

  If "A" and "B" each has an encrypted connection to a third party "C", "C" can deliver a key on the encrypted links to "A" and "B".
- **Required Keys of Symmetric Key Distribution**

$$ {}_{N}C_{2} = \frac{N(N-1)}{2} = O(N^2) $$

- **Centralized Key Distribution Scenario**

  Initiator "A", Responder "B",

  Key Distribution Center (KDC)

  Random Number used only once (Nonce): N_i

  1. A to KDC

$$ \mathrm{ID_A} \parallel \mathrm{ID_B} \parallel N_1 $$

  2. KDC to A

$$ E(K_A, [K_S \parallel \mathrm{ID_A} \parallel \mathrm{ID_B} \parallel N_1]) \parallel E(K_b, [K_S \parallel \mathrm{ID_A}]) $$

  3. A to B

$$ E(K_b, [K_S \parallel \mathrm{ID_A}]) $$

  4. B to A

$$ E(K_S, N_2) $$

  5. A to B

$$ E(K_S, f(N_2)) $$

- **Hierarchical Key Control**

  For communication among entities within the same local domain, the local KDC is responsible for key distribution.

  If two entities in different domains desire a shared key, then the corresponding local KDC's can communicate through a global KDC.

  This scheme minimizes the effort involved in master key distribution because most master keys are those shared by a local KDC with its local entities. It limits the range of a faulty or subverted KDC to its local area only.

- **Session Key Lifetime**

  The more frequently session keys are exchanged, the more secure they are, but, the distribution of session keys delays the start of any exchange and places a burden on network capacity.

  For connection-oriented protocols, one choice is to use the same session key for the length of time that the connection is open, using a new session key for each new session.

  For a connectionless protocol, there is no explicit connection, thus it is not obvious how often one needs to change the session key.

- **Decentralized Key Distribution**

  Initiator "A", Responder "B",
  Random Number used only once (Nonce): N_i

  1. A to B

  $$\mathrm{ID_A} \parallel N_1$$

  2. B to A

  $$E(K_m, [K_S \parallel \mathrm{ID_A} \parallel \mathrm{ID_B} \parallel f(N_1) \parallel N_2])$$

  3. A to B

  $$E(K_S, f(N_2))$$

- **Public-Key Distribution of Secret Keys**

  Initiator "A", Responder "B",
  Random Number used only once (Nonce): N_i

  1. A to B

  $$E(\mathrm{PU_B}, [N_1 \parallel \mathrm{ID_A}])$$

  2. B to A

  $$E(\mathrm{PU_A}, [N_1 \parallel N_2])$$

  3. A to B

  $$E(\mathrm{PU_B}, N_2)$$

  $$E\big(\mathrm{PU_B}, E(\mathrm{PR_A}, K_S)\big)$$

- **Public Announcement**

  Although this approach is convenient, it has a major weakness. Anyone can forge such a public announcement. That is, some user could pretend to be user A and send a public key to another participant or broadcast such a public key. Until such time as user A discovers the forgery and alerts other participants, the forger is able to read all encrypted messages intended for A and can use the forged keys for authentication.

  공개된 키를 이용한 신분 위조 가능성이 발생함. 사용자 A가 위조를 발견하고 다른 참가자에게 경고할 때까지 위조자는 A용으로 암호화 된 모든 메시지를 읽을 수 있으며 인증용 위조 키를 사용할 수 있음.

- **Publicly Available Directory**

  A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys. Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organization.

  This scheme would include the following elements.

  1. The authority maintains a directory with a {name, public key} entry for each participant.
  2. Each participant registers a public key with the directory authority. Registration would have to be in person or by some form of secure authenticated communication.
  3. A participant may replace the existing key with a new one at any time, either because of the desire to replace a public key that has already been used for a large amount of data, or because the corresponding private key has been compromised in some way.
  4. Authenticated communication from the authority to the participant is mandatory.

  디렉토리 권한이 해킹 가능성이 있음.

- **Public-Key Distribution Scenario**

Initiator "A", Responder "B",

Public-Key Authority (Auth)

Random Number used only once (Nonce): N_i

1. A to Auth

$$\text{Request} \parallel T_1$$

2. Auto to A

$$E(\text{PR}_{\text{Auth}}, [\text{PU}_{\text{B}} \parallel \text{Request} \parallel T_1])$$

3. A to B

$$E(\text{PU}_{\text{B}}, [\text{ID}_{\text{A}} \parallel N_1])$$

4. B to Auth

$$\text{Request} \parallel T_2$$

5. Auth to A

$$E(\text{PR}_{\text{Auth}}, [\text{PU}_{\text{A}} \parallel \text{Request} \parallel T_2])$$

6. B to A

$$E(\text{PU}_{\text{A}}, [N_1 \parallel N_2])$$

7. A to B

$$E(\text{PU}_{\text{A}}, N_2)$$

- **Public-Key Certificates**

Initiator "A", Responder "B",

Certificate Authority (CA)

1. A to CA

$$\text{PU}_{\text{B}}$$

2. CA to A

$$C_A = E(\text{PR}_{\text{Auth}}, [T_1 \parallel \text{ID}_{\text{A}} \parallel \text{PU}_{\text{A}}])$$

3. B to CA

$$\text{PU}_{\text{B}}$$

4. CA to B

$$C_B = E(\text{PR}_{\text{Auth}}, [T_2 \parallel \text{ID}_{\text{B}} \parallel \text{PU}_{\text{B}}])$$

-