Key Management Protocol for IPsec

Diffie–Hellman key exchange algorithm

- Provides a mechanism that allows two users to agree on a shared secret key without requiring encryption
  - $Yi \equiv \alpha^{Xi} \pmod q$
  - $Yj \equiv \alpha^{Xj} \pmod q$
  - $Kij \equiv \alpha^{XiXj} \pmod q$
- Kij is common secret key
- This shared key is immediately available for use in encrypting subsequent data transmission
- Allows two users to agree on a shared secret key without requiring encryption

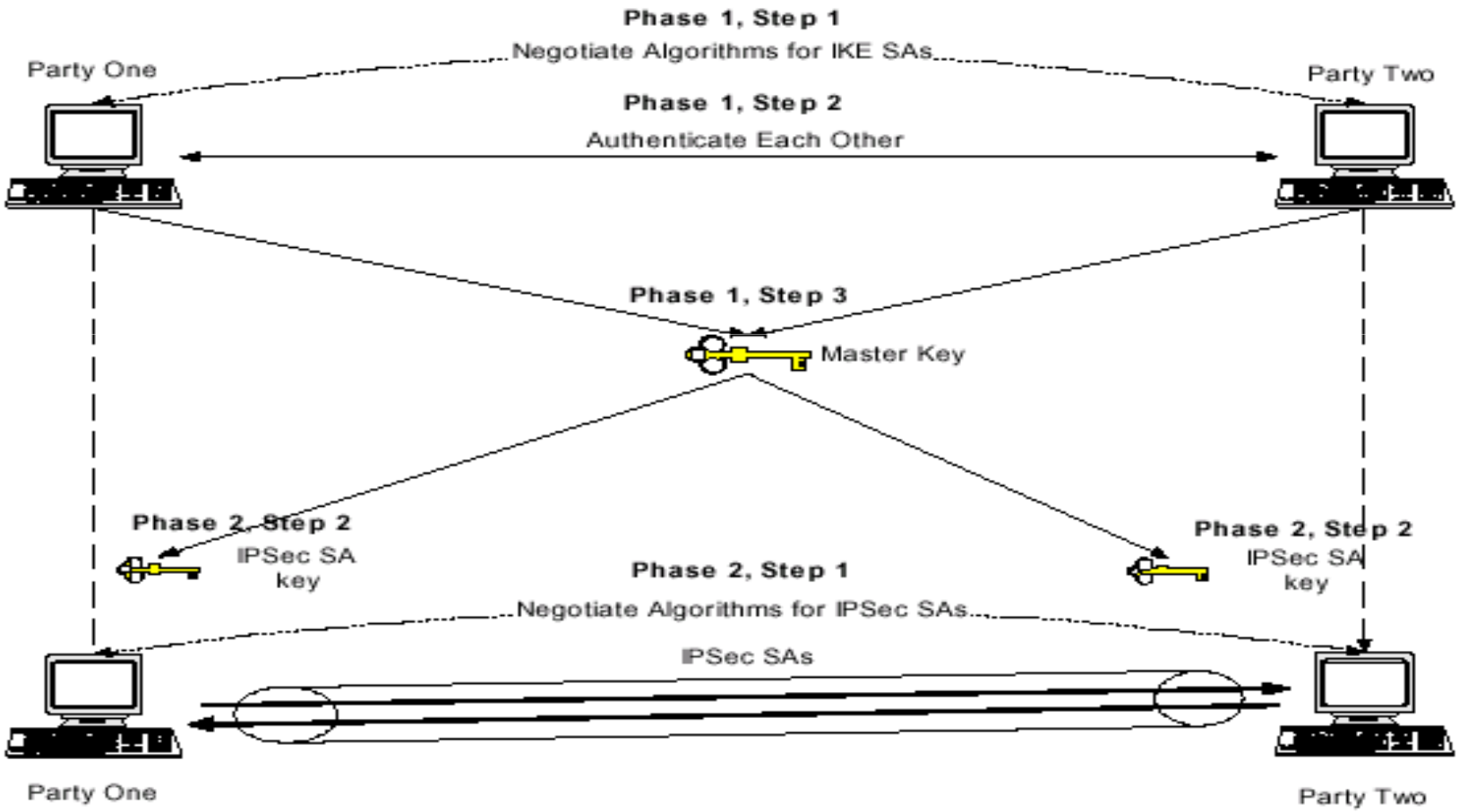
Key management

- Two types of key establishment
  - Manual
    - System administrator configures each system with the necessary keys
  - Automated
    - On-demand creation of keys for SA
- Key establishment
  - Secure key distribution is an essential part and the heart of data protection
  - Relies on cryptography
  - An automated protocol makes the process feasible on the Internet
  - This automated process is the IKE
  - IKE = ISAKMP + OAKELY key exchange

Key Management Protocol for IPSec

- IPSec need to
  - Set up the SA before establishing a secure session
  - Negotiate the terms that are defined in the SA
  - Determine and distribute secret key
  - Then establish a session key
- Default automated method is ISAKMP/Oakley
  - Oakley key determination protocol
    - A key exchange protocol based on Diffie-Hellman
    - Provides added security (e.g., authentication)
  - ISAKMP – Internet Security Association and Key Management Protocol
    - Provides a framework for key exchange
    - Defines message formats that can carry the messages of various key exchange protocols

Key Management



OAKLEY Key Determination Protocol

- Key agreement protocol
- A refinement of the Diffie–Hellman key exchange algorithm
- Helps authenticated parties to exchange keying material across an insecure connection
- Can be used directly over the IP protocol or over UDP protocol using a well-known port number assignment available

OAKLEY Key Determination Protocol

- Cookie
  - Fast hash (e.g. MD5)
    - Cookie = H(IP Src add, IP Dest add, UDP src &Dest port, locally generated secret random value)
  - Cookie should be unique for each SA
    - Therefore, the date and time MUST be added to the information hashed
    - Oakley employs nonces to ensure against replay attacks
    - Each nonce is a pseudorandom number which is generated by the transmitting entity
    - The nonce payload contains this random data used to guarantee liveliness during a key exchange and protect against replay attacks

OAKLEY Key Determination Protocol

- Purposes:
  - Act as anti-clogging token
    - Protect from denial of service
    - Provide a form of source address identification for both parties
  - To prevent replay attacks
  - Prevents an attacker from obtain a cookie using a real IP address and UDP port
  - Protect the computing resources from attack without spending excessive CPU resources to determine its authenticity
  - Key naming

- All the Oakley message fields correspond to ISAKMP message payloads

ISAKMP

- RFC 2408
- Establish SA and cryptographic keys
- Provides only a Framework for SA management and cryptographic key establishment
- Support the negotiation of SAs for security protocols at all layers of the network stack
- Defines procedures and packet formats to establish, negotiate, modify and delete SA
- It is independent of the key generation technique, encryption algorithm and authentication mechanism
- Protocols such as Internet Key Exchange provide authenticated keying material for use with ISAKMP

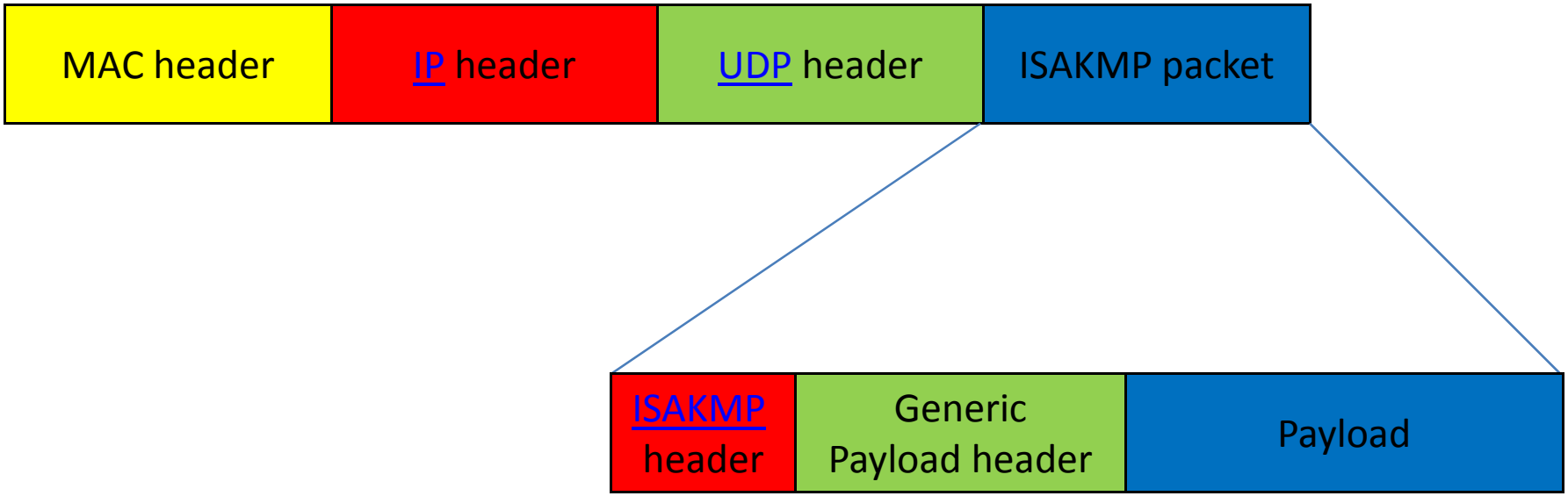
ISAKMP

- It defines payloads for exchanging key generation and authentication data
- These payload formats provide a consistent framework for transferring key and authentication data

- By centralising the management of the SAs, ISAKMP reduces the amount of duplicated functionality within each security protocol

(I) ISAKMP Payloads

- Provide modular building blocks for constructing ISAKMP messages
- The presence and ordering of payloads in ISAKMP is defined the Exchange Type Field in ISAKMP Header



(I) ISAKMP Payloads

- **ISAKMP Header**

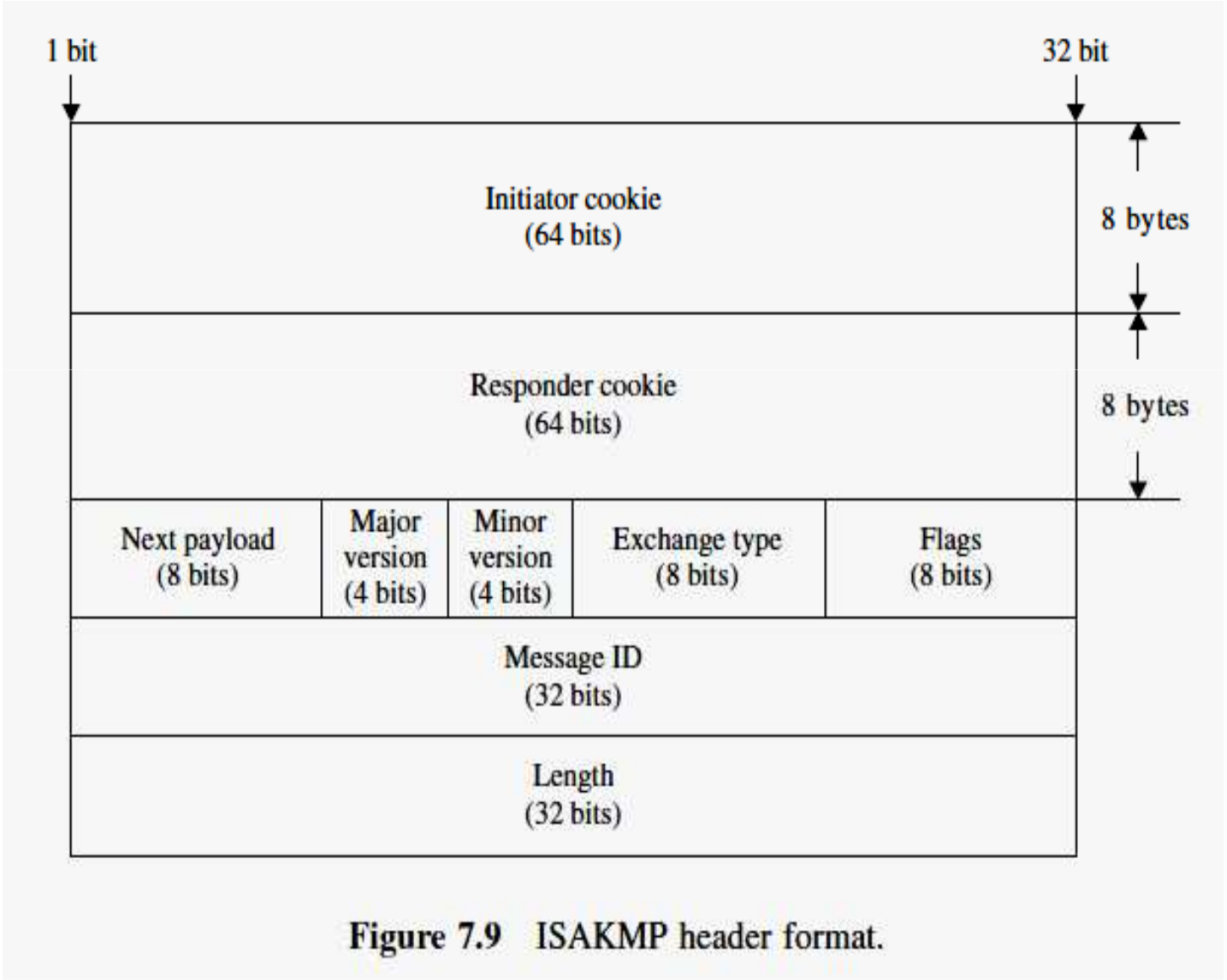


Figure 7.9 ISAKMP header format.

(I) ISAKMP Payloads

ISAKMP Header

- **Initiator Cookie (64 bits)**
  - Cookie of entity that initiated SA establishment, notification or SA deletion
- **Responder Cookie (64 bits)**
  - Cookie of entity that is corresponded to an SA establishment request, notification or deletion
- **Next Payload (8 bits)**
  - This field indicates the type of the first payload in the message

Next Payload Type	Value
NONE	0
Security Association (SA)	1
Proposal (P)	2
Transform (T)	3
Key Exchange (KE)	4
Identification (ID)	5
Certificate (CERT)	6
Certificate Request (CR)	7
Hash (HASH)	8
Signature (SIG)	9
Nonce (NONCE)	10
Notification (N)	11
Delete (D)	12
Vendor ID (VID)	13
RESERVED	14 - 127
Private USE	128 - 255

(I) ISAKMP Payloads

ISAKMP Header

- **Major Version (4 bits)**
  - Indicates the Major version of the ISAKMP protocol in use
  - Set the Major version to 1 according to ISAKMP Internet-Draft
- **Minor Version (4 bits)**
  - Indicates the Minor version of ISAKMP protocol in use
  - Set the Minor version to 0 according to implementations based on the ISAKMP Internet-Draft
-

(I) ISAKMP Payloads

ISAKMP Header

- **Exchange Type (8 bits)**
  - Indicates the type of exchange being used
  - This dictates the message and payload orderings in the ISAKMP exchanges
  - Exchange Type Value
    - RESERVED 0-33
    - IKE_SA_INIT 34
    - IKE_AUTH 35
    - CREATE_CHILD_SA 36
    - INFORMATIONAL 37
    - RESERVED TO IANA 38-239
    - Reserved for private use 240-255

ISAKMP

ISAKMP Header

- **Flags (8 bits)**
  - Indicates specific options that are set for the ISAKMP exchange
  - Beginning with LSB :
    - Bit 0 - Encryption bit
    - Bit 1- Commit bit
    - Bit 2 - Authentication only bit
    - The remaining bits of the Flags field must be set to 0 prior to transmission

ISAKMP

ISAKMP Header

- **Flags (8 bits)**
  - Encryption Bit
    - If set (1), all payloads following the header are encrypted using the encryption algorithm identified in the ISAKMP SA
      - ISAKMP SA Identifier = initiator cookie and responder cookie
    - If not set (0), the payloads are not encrypted
  - Commit Bit
    - This bit is used to signal key exchange synchronization
    - It is used to ensure that encrypted material is not received prior to completion of the SA establishment
    - Protect against loss of transmissions over unreliable networks
    - Guard against the need for multiple retransmissions
  - Authentication Only Bit
    - This bit is intended for use with the Informational Exchange with a Notify payload
    - Allow the transmission of information with integrity checking, but no encryption

ISAKMP

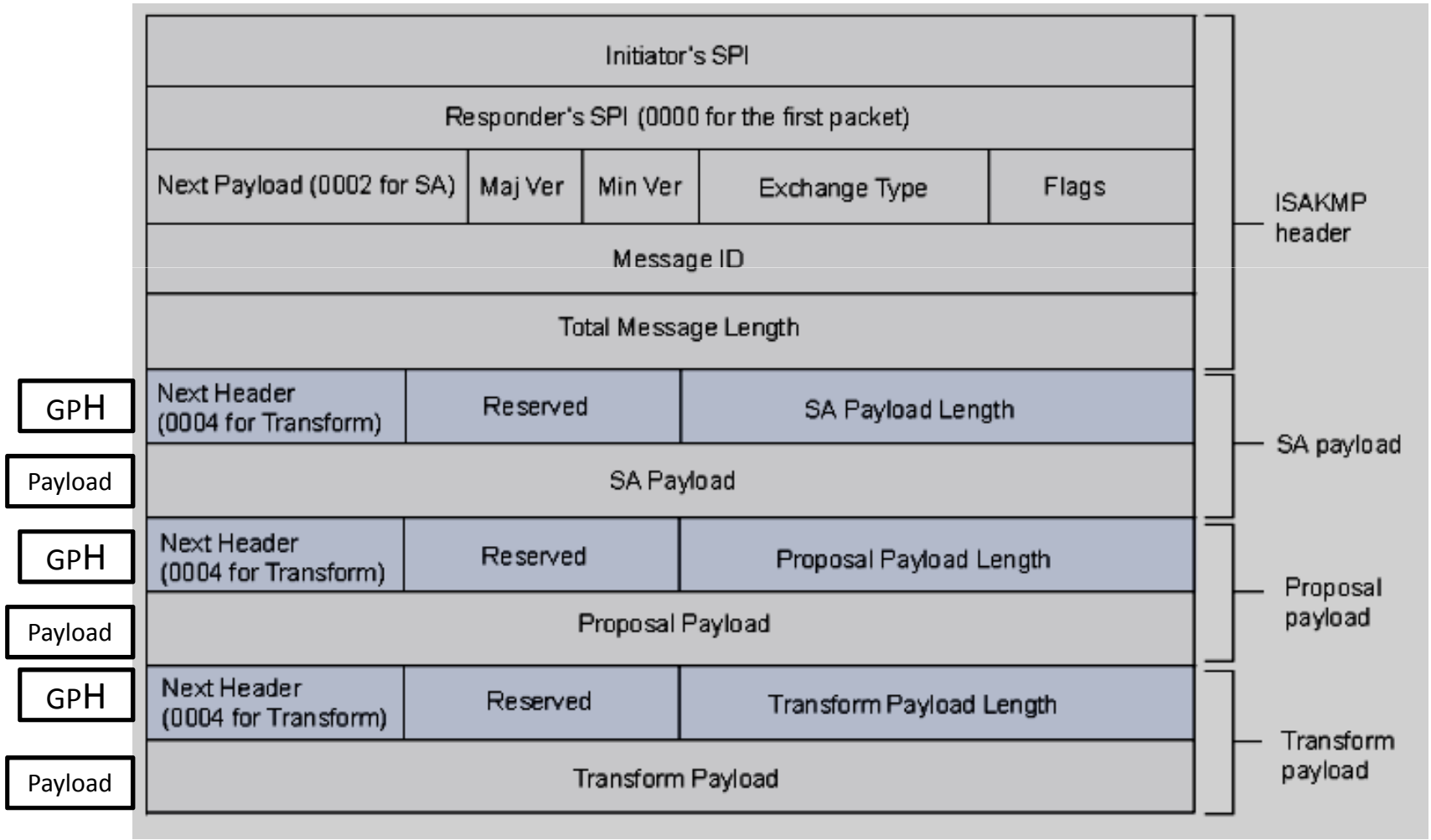
ISAKMP Header

- **Message ID (32 bits)**
  - During Phase 1 negotiation, this value must be set to 0
  - Message ID is used to identify protocol state during Phase 2 negotiations
  - This value is randomly generated by the initiator of the phase 2 negotiation
  - Differentiate SA during simultaneous SA establishments
  - Used to control retransmission of lost packets and matching requests and responses

- **Length (32 bits)**
  - Length of total message (header || payload) is 32 bits

ISAKMP

SAKMP Header with Generic ISAKMP Payloads



ISAKMP

- **Generic Payload Header**
  - Each ISAKMP payload begins with a generic header
  - Provides a payload chaining capability
  - Clearly defines the boundaries of a payload

  - **Next Payload (8 bits)**
    - Identifier for the payload type of the next payload in the message
    - If the current payload is the last in the message, then this field will be 0
    - This field provides the chaining capability
  - **Reserved (8 bits)**
    - This field is not used and set to 0
  - **Payload Length (16 bits)**
    - Indicates the length in bytes of the current payload, including the generic payload header

ISAKMP

- **Payload Types for ISAKMP**
- Payloads are used to transfer information such as SA data or key exchange data in DOI-defined formats
    1. **SA :** used to begin the setup of a new SA; carries various attributes
    2. **Proposal (P):** used during SA setup; indicates protocol to be used (AH or ESP) and number of transforms
    3. **Transform (T) :** used during SA setup; indicates transform (e.g., DES, 3DES) and its attributes
    4. **IKE :** used to carry key exchange data (e.g., Oakley)
    5. **Identification (ID) :** used to exchange identification information (e.g., IP address)
    6. **Certificate Payload :** carries a public key certificate (PGP, X.509, SPKI, …)
    7. **Certificate Request Payload**
    8. **Hash (HASH)**
    9. **Signature Payload**
    10. **Nonce (NONCE)**
    11. **Notification (N) :** contains error or status information
    12. **Delete Payload :** indicates one or more SAs that the sender has deleted from its database (no longer valid)
    13. **Vendor ID**

ISAKMP

- **Security Association Payload**
  - Used to negotiate security attributes
  - The Security Association Payloads are defined as follows:
    - The Next Payload field (8 bits)
      - This field has a value of 0 if this is the last payload in the message
      - Otherwise 1 - IPSEC
    - The Reserved field (8 bits)
      - unused, set to 0
    - The Payload Length field (16 bits)
      - including the SA payload, all Proposal payloads, and all Transform payloads associated with the proposed SA
    - The Situation field (Payload) (variable length)
      - Defines policy decisions regarding the security attributes being negotiated

ISAKMP

- **Security Association Payload**

Next payload	Reserved	Payload length
Situation field		

  - An SA payload consists of one or more proposals
  - Each proposal includes one protocol
  - Each protocol contains one or more transforms
  - Altogether each proposal specifying a cryptographic algorithm

ISAKMP

- **Proposal Payload**
  - The payload type for the Proposal Payload is two(2)
  - The Proposal Payload fields are defined as follows:
    - The Next Payload field (8 bits)
      - Indentifies the payload type of the next payload in the message
      - This field must only contain the value 2 or 0
        - » 2 for additional Proposal Payloads in the message
        - » 0 when the current Proposal Payload is the last within the SA proposal
    - The Reserved field (8 bits)
      - Set to 0 and is reserved it for the future use
    - The Payload Length field (16 bits)
      - Length of the entire Proposal payload, including generic payload header, the Proposal Payload, and all Transform payloads associated with this proposal

ISAKMP

- **Proposal Payload**
  - The Proposal Num field (8 bits)
    - Identifies the proposal number for the current payload
  - The Protocol-id field (8 bits)
    - Specifies the protocol identifier for the current negotiation

      Examples might include IPsec ESP, IPsec AH, TLS, etc.
  - The SPI Size (8 bits)
    - Denotes the length of SPI
    - In the case of ISAKMP, the Initiator and Responder cookie pair from the ISAKMP Header is the ISAKMP SPI
    - The SPI size may be from zero(0) to sixteen (16)
  - Transform (8 bits)
    - Specifies the number of transforms for the proposal, each of these is contained in a Transform Payload

ISAKMP

- **Proposal Payload**

0 or 2	Reserved	Proposal length	
Proposal num	Protocol	SPI size	Num transform
SPI			
<Transform>			

ISAKMP

- **Transform Substructure**

0 or 3	Reserved	Transform length
Transform Type	Reserved	Transform ID
Transform Attributes		

Transform Type

	Transform Type	Used In
RESERVED	0	
Encryption Algorithm (ENCR)	1	(IKE and ESP)
Pseudo-random Function (PRF)	2	(IKE)
Integrity Algorithm (INTEG)	3	(IKE, AH, optional in ESP)
Diffie-Hellman Group (D-H)	4	(IKE, optional in AH & ESP)
Extended Sequence Numbers (ESN)	5	(AH and ESP)
RESERVED TO IANA	6-240	
PRIVATE USE	241-255	

Transform ID for Transform Type 1 and 2

Transform Type 1 (Encryption Algorithm),

Name	Number
RESERVED	0
ENCR_DES_IV64	1
ENCR_DES	2
ENCR_3DES	3
ENCR_RC5	4
ENCR_IDEA	5
ENCR_CAST	6
ENCR_BLOWFISH	7
ENCR_3IDEA	8

Transform Type 2 (Pseudo-random Function

Name	Number
RESERVED	0
PRF_HMAC_MD5	1
PRF_HMAC_SHA1	2
PRF_HMAC_TIGER	3
PRF_AES128_XCBC	4

ISAKMP

- **Transform Payload**
  - Specific security mechanism to be used to secure the communications channel
  - Present several possible supported transforms for the proposed protocol
  - The payload type for the Transform Payload is three (3)
  - The Transform Payload field s are defined as follows:
    - The Next Payload field (8 bits)
      - Identifies the payload type of the next payload in the message
    - This field must only contain the value 3 or 0
      - 3 - There are additional Transform payloads in the proposal
      - 0 - when the current Transform Payload is the last within the proposal

ISAKMP

- **Transform Payload**
  - The Reserved field (8 bits)
    - Unused, set to 0
  - The Transform Num field (8 bits)
    - identifies the Transform number for the current payload
  - The Transform-id field (8 bits)
    - specifies the Transform identifier for the protocol within the current proposal
  - The Reserved 2 field (16 bits)
    - unused, set to 0

ISAKMP

- **Transform Payload**
  - The SA Attributes field (variable length)
    - Contains the security association (SA) attributes
    - The SA Attributes should be represented using the Data Attributes format.
      - Type/Length/Value (TLV) when AF=0
      - Type/Value (TV) when AF=1

Data Attributes format

AF	Attribute Type	Attribute length / Value When AF= 0/1
Attribute Value/ Not Transmitted When AF= 0/1		

Attribute Types

class	value
SA Life Type	1
SA Life Duration	2
Group Description	3
Encapsulation Mode	4
Authentication Algorithm	5
Key Length	6
Key Rounds	7
Compress Dictionary Size	8
Compress Private Algorithm	9

ISAKMP

- **Key Exchange Payload**
  - The Key Exchange Payload supports a variety of key exchange techniques
  - Example key exchanges are Oakley, Diffie-Hellman, the enhanced D-H key exchange, and the RSA-based key exchange used by PGP
  - The Key Exchange Payload fields are defined as follows:
    - The Next Payload field (8 bits)
    - The Reserved field (8 bits)
    - The Payload Length field (16 bits)
    - The Key Exchange Data field (variable length)
      - Data required to generate a session key

Next payload	Reserved	Payload length
Key Exchange Data		

ISAKMP

- **Identification Payload**
  - Used for determining the identities of communication partners (IPv4,v6, email address)
  - May be used for determining authenticity of information
  - The payload type for the Identification Payload is five(5)
  - The Identification Payload fields are described as follows:
    - The Next Payload field (8 bits), The Reserved field (8 bits), The Payload Length field (16 bits)
    - The ID type field (8 bits)
      - Specifies the type of identification being used
    - The DOI specific ID Data field (24 bits)
      - Contains DOI specific identification data. If unused, set to 0
    - The Identification Data field (variable length)
      - Contains identity information

Next payload	Reserved	Payload length
ID Type	DOI Specific ID Data	
Identification Data		

ISAKMP

- **Certificate Payload**
  - Transport certificates via ISAKMP
  - Used when directory service is not available to distribute certificates
  - The Payload type for the Certificate payload is six(6)
  - The Certificate Payload fields are defined as follows:
    - The Next Payload field (8 bits, The Reserved field (8 bits), The Payload Length field (16 bits)
    - The Certificate Encoding field (8 bits)
      - Indicates the type of certificate
    - The Certificate Data field (variable length)
      - denotes actual encoding of certificate data

<table>
<tr><td>Next payload</td><td>Reserved</td><td>Payload length</td></tr>
<tr><td>Cert. Encoding</td><td colspan="2" rowspan="2"></td></tr>
<tr><td colspan="2">Certificate data</td></tr>
</table>

ISAKMP

- **Certificate Request Payload**
  - Provides a mean to request certificate
  - The payload type for the Certificate Request Payload is seven(7).
  - The Certificate Request Payload fields are defined as follows:
    - The Next Payload field (8 bits), The Reserved field (8 bits), The Payload Length field (16 bits)
    - The Certificate Type/ Encoding field (8 bits)
    - The Certificate Authority field (variable length)

Next payload	Reserved	Payload length
Cert. Type		
Certificate Authority		

Certificate Encoding	Value
X.509 Certificate - Signature	4
Raw Public Key	15

ISAKMP

- **Hash Payload**
  - Contains data generated by the hash function over some part of the message
  - Used to verify the integrity of the data in an ISAKMP message or for authentication of the negotiating entities
  - The payload type for the Hash Payload is eight(8)
  - The Hash Payload fields are defined as follows:
    - The Next Payload field (8 bits), The Reserved field (8 bits), The Payload Length field (16 bits)
    - The Hash Data field (variable length)

Next payload	**Reserved**	**Payload length**
Hash Data		

ISAKMP

- **Signature Payload**
  - Contains data generated by the digital signature function, over some part of the message
  - This payload is used to verify the integrity of the data in the ISAKMP message, and may be of use for non-repudiation services.
  - The payload type for the Signature Payload is nine(9).
  - The Signature Payload fields are defined as follows:
    - The Next Payload field (8 bits), The Reserved field (8 bits), The Payload Length field (16 bits)
    - The Signature Data field (variable length)
      - Data that results from applying the digital signature function to the ISAKMP message and/or state

Next payload	**Reserved**	**Payload length**
Signature Data		

ISAKMP

- **Nonce Payload**
  - Contains random data used to guarantee liveness during an exchange and protect against replay attacks
  - The nonces may be transmitted as part of the key exchange data, or as a separate payload
  - The Payload type for the Nonce Payload is ten(10)
  - The Nonce Payload fields are defined as follows:
    - The Next Payload field (8 bits) ,The Reserved field (8 bits), The Payload Length field (16 bits)
    - The Nonce Data field (variable length)
      - Contains the random data generated by the transmitting entity

Next payload	**Reserved**	**Payload length**
Nonce Data		

ISAKMP

- **Notification Payload**
  - Transmit information data, such as error conditions
  - Possible to send multiple Notification Payloads in a single ISAKMP message
  - The payload type for the Notification Payload is eleven (11).
  - The Notification Payload fields are defined as follows:
    - The Next Payload field (8 bits), The Reserved field (8 bits), The Payload Length field (16 bits)

Next payload	Reserved	Payload length
DOI		
Protocol ID	SPI size	Notify Message Type
SPI		
Notification Data		

ISAKMP

- **Notification Payload**
  - DOI (32 bits)
    - identifies the DOI
      - » ISAKMP DOI value is zero(0)
      - » IPsec DOI it is one (1)
  - The Protocol-id field (8 bits)
  - The SPI Size field (8 bits)
  - The Notify Message Type field (16 bits)
    - Specifies the type of notification message
  - The Security Parameter Index (SPI)
  - The Notification Data field (variable length)
    - Informational or error data

ISAKMP

- **Delete Payload**
  - Contains a SA that the sender has removed from its SA database
  - The Payload type for the Delete Payload is twelve(12)
  - The Delete Payload fields are defined as follows:
    - The Next Payload field (8 bits), The Reserved field (8 bits) is unused, The Payload Length field (16 bits)
    - The Domain of Interpretation field (32 bits)
    - The Protocol-id field (8 bits)
    - The SPI Size field (8 bits)
    - The Num of SPIs field (16 bits)
      - the number of SPIs contained in the Delete Payload
    - The Security Parameter Indexes field (variable length)
      - Identifies the specific security associations to delete

ISAKMP

- **Delete Payload**

Next payload	Reserved	Payload length
DOI		
Protocol ID	SPI size	Number of SPI
SPI		

ISAKMP

- **Vendor ID Payload**
  - Contains vendor defined constant
  - This mechanism allows a vendor to experiment with new features while maintaining backwards compatibility
  - The Payload type for the Vendor ID Payload is thirteen(13)
    - The Vendor ID Payload fields are defined as follows:
    - The Next Payload field (8 bits), The Reserved field (8 bits), The Payload Length field (16 bits)
    - The Vendor ID field (variable length)
      - » keyless hash of a string containing the product name, and the version of the product

ISAKMP

- **Vendor ID Payload**

Next payload	Reserved	Payload length
Vendor ID		

ISAKMP

- **(III) ISAKMP Exchanges**
  - Exchanges define the **content and ordering of ISAKMP messages** during communications between peers
  - Most exchanges will include all the basic payload types - SA, KE, ID, SIG - and may include others
  - The primary difference between exchange types is the ordering of the messages and the payload ordering within each message
  - While the ordering of payloads within messages is not mandated, for processing efficiency it is recommended that the SA payload be the first payload within an exchange

ISAKMP

- **ISAKMP Exchanges**

  - **Base Exchange**
    - Allow the Key Exchange and Authentication-related information to be transmitted together
    - Combining the Key Exchange and Authentication related information into one message reduces the number of round-trips at the expense of not providing identity protection

  - **Identity Protection Exchange**
    - Designed to separate the Key Exchange information from the Identity and Authentication-related information
    - Provides protection of the communicating identities at the expense of two additional messages
    - Identities are exchanged under the protection of a previously established common shared secret

ISAKMP

- **ISAKMP Exchanges**
  - **Authentication Only Exchange**
    - Allow only Authentication-related information to be transmitted
    - None of the transmitted information will be encrypted
    - But the authentication only exchange will be encrypted by the ISAKMP SA, negotiated in the first phase
  - **Aggressive Exchange**
    - Allow the Security Association, Key Exchange and Authentication-related payloads to be transmitted together
    - Reduces the number of round-trips at the expense of not providing identity protection

ISAKMP

- **ISAKMP Exchanges**
  - **Informational Exchange**
    - One-way transmittal of information that can be used for security association management
    - If the Informational Exchange occurs
      - Prior to the exchange of keying material during an ISAKMP Phase 1 negotiation, there will be no protection provided for the Information Exchange
      - Once keying material has been exchanged or an ISAKMP SA has been established, the Informational Exchange must be transmitted under the protection provided by the keying material or the ISAKMP SA

ISAKMP

(IV) ISAKMP Payload Processing

– General Message Processing

- Basic processing applied to insure protocol reliability and to minimize threats such as denial of services and replay attacks
- All processing should include **packet length checks** to insure the packet received is at least as long as the length given in the ISAKMP Header
- If the ISAKMP message length and the value in the Payload Length field of the ISAKMP Header are not the same, then ISAKMP message must be rejected

ISAKMP

- **ISAKMP Payload Processing**
  - **ISAKMP Header Processing**
    - Initiator (transmitter) must create the respective cookie, determine the relevant security characteristics of the session, construct an ISAKMP Header with fields, and transmit the message to the destination host (responder).
    - The responder (receiver) must verify the Initiator and Responder cookies, check the Next Payload field to confirm it is valid, check the Major and Minor Version fields to confirm they are correct, check the Exchange Type field to confirm it is valid, check the Flags field to ensure it contains correct values, and check the Message ID field to ensure it contains correct values.
    - Thus, processing of the ISAKMP message continues using the value in the Next Payload field.

ISAKMP

- **ISAKMP Payload Processing**
  - **Generic Payload Header Processing**
    - Initiator must place the value of the Next Payload in the Next Payload field, place the value zero(0) in the Reserved field, place the length (in octets) of the payload in the Payload Length field, and construct the payloads
    - Responder must check the Next Payload field to confirm it is valid, verify the Reserved field contains the value zero(0), and process the remaining payloads as defined by the Next Payload field

ISAKMP

- **ISAKMP Payload Processing**
  - **Security Association Payload Processing**
    - Initiator must determine the DOI for which this negotiation is being preformed, determine the situation, determine the proposal(s) and transform(s) within the situation,
    - Construct a SA payload, and transmit the message to the receiving entity (responder).
    - Responder must determine if the DOI is supported
    - Process the remaining payloads (Proposal, Transform) of the SA payload.
    - If the SA Proposal is not accepted, then the Invalid Proposal event may be logged in the appropriate system audit file
    - An Information Exchange with a Notification can be send

ISAKMP

- **ISAKMP Payload Processing**
  - **Proposal Payload Processing**
    - Initiator must determine the Protocol for this proposal, determine the number of proposals to be offered for this proposal and the number of transform for each proposal, generate a unique pseudo-random SPI, and construct a Proposal payload
    - When a Proposal payload is received, the receiving entity (responder) must determine if the proposal is supported and if the Protocol-ID field is invalid, determine whether the SPI is valid or not, ensure whether or not proposals are formed correctly, and then process the Proposal and Transform payloads as defined by the Next Payload field

ISAKMP

- **ISAKMP Payload Processing**
  - **Transform Payload Processing**
    - When creating a Transform Payload, the transmitting entity (initiator) must determine the Transform num for this transform, determine the number of transforms to be offered for this proposal, and construct a Transform payload
    - When a Transform payload is received, the receiving entity responder) must do as follows: Determine if the Transform is supported. If the Transform-ID field contains an unknown or unsupported value, then that Transform payload must be ignored
    - Ensure Transforms are presented according to the details given in the Transform Payload and Security Association establishment. Finally, process the subsequent Transform and Proposal payloads as defined by the Next Payload field.

ISAKMP

- **ISAKMP Payload Processing**
- **Key Exchange Payload Processing**
  - When creating a Key Exchange payload, the transmitting entity (initiator) must determine the Key Exchange to be used as defined by the DOI, determine the usage of Key Exchange Data field as defined by the DOI, and construct a Key Exchange payload. Finally, transmit the message to the receiving entity (responder).
  - When a Key Exchange payload is received, the receiving entity (responder) must determine if the Key Exchange is supported.
  - If the Key Exchange determination fails, the message is discarded and the following actions are taken:
    - The event of Invalid Key Information may be logged in the appropriate system audit file.
    - An Informational Exchange with a Notification payload containing the Invalid-Key- Information message type may be sent to the transmitting entity.
    - This action is dictated by a system security policy.

ISAKMP

- **ISAKMP Payload Processing**
  - **Identification Payload Processing**
    - When an Identification Payload is created, the transmitting entity (initiator) must determine the Identification information to be used as defined by the DOI, determine the usage of the Identification Data field as defined by the DOI, construct an Identification payload, and finally transmit the message to the receiving entity
    - When an Identification payload is received, the receiving entity (responder) must determine if the Identification Type is supported.
    - This may be based on the DOI and Situation.
    - If the Identification determination fails, the message is discarded.
    - An Informational Exchange with a Notification payload containing the Invalid-ID-Information message type is sent to the transmitting entity (initiator)

ISAKMP

- **ISAKMP Payload Processing**
  - **Certificate Payload Processing**
    - When a Certificate Payload is created, the transmitting entity (initiator) must determine the Certificate Encoding which is specified by the DOI, ensure the existence of a certificate formatted as defined by the Certificate Encoding, construct a Certificate payload, and then transmit the message to the receiving entity (responder)
    - When a Certificate payload is received, the receiving entity (responder) must determine if the Certificate Encoding is supported
    - If the Certificate Encoding is not supported, the payload is discarded
    - The responder then process the Certificate Data field. If the Certificate Data is improperly formatted, the payload is discarded

ISAKMP

- **ISAKMP Payload Processing**
  - **Certificate Request Payload Processing**
    - When creating a Certificate Request Payload, the transmitting entity (initiator) must determine the type of Certificate Encoding to be requested, determine the name of an acceptable Certificate Authority, construct a Certificate Request payload, and then transmit the message to the receiving entity (responder).
    - When a Certificate Request payload is received, the receiving entity (responder) must determine if the Certificate Encoding is supported. If the Certificate Encoding is invalid, the payload is discarded. The responder must determine if the Certificate Authority is supported for the specified Certificate Encoding. If the Certificate Authority is improperly formatted, the payload is discarded. Finally, the responder must process the Certificate Request. If a requested Certificate Type with the specified Certificate Authority is not available, then the payload is discarded.

ISAKMP

- **ISAKMP Payload Processing**
  - **Hash Payload Processing**
    - When creating a Hash Payload, the transmitting entity (initiator) must determine the Hash function to be used as defined by the SA negotiation, determine the usage of the Hash Data field as defined by the DOI, construct a Hash payload, and then transmit the message to the receiving entity (responder).
    - When a Hash Payload is received, the receiving entity (responder) must determine if the Hash is supported. If the Hash determination fails, the message is discarded. The responder also performs the Hash function as outlined in the DOI and/or Key Exchange protocol documents. If the Hash function fails, the message is discarded.

ISAKMP

- **ISAKMP Payload Processing**
  - **Signature Payload Processing**
    - When a Signature Payload is created, the transmitting entity(initiator) must determine the Signature function to be used as defined by the SA negotiation, determine the usage of the Signature Data filed as defined by the DOI, construct a Signature payload, and finally transmit the message to the receiving entity (responder).
    - When a Signature payload is received, the receiving entity must determine if the Signature is supported. If the Signature determination fails, the message is discarded. The responder must perform the Signature function as outlined in the DOI and/or Key Exchange protocol documents. If the Signature function fails, the message is

ISAKMP

- **ISAKMP Payload Processing**
  - **Nonce Payload Processing**
    - When creating a Nonce Payload, the transmitting entity (initiator) must create a unique random values to be used as a nonce, construct a Nonce payload, and transmit the message to the receiving entity.
    - When a Nonce Payload is received, the receiving entity (responder) must do as follows: There are no specific procedures for handling Nonce payloads. The procedures are defined by the exchange types and possibly the DOI and Key Exchange descriptions.

ISAKMP

- **ISAKMP Payload Processing**
  - **Notification Payload Processing**
    - During communications it is possible that errors may occur.
    - The Information Exchange with a Notify Payload provides a controlled method of informing a peer entity that occur has occurred during protocol processing.
    - It is recommended that Notify Payloads be sent in a separate Information Exchange rather than appending a Notify Payload to an existing exchange.

ISAKMP

- **ISAKMP Payload Processing**
  - **Notification Payload Processing**
    - When a Notification Payload is created, the transmitting entity (initiator) must determine the DOI for this Notification, determine the Protocol-ID for this Notification, determine the SPI size based on the Protocol-ID field, determine the Notify Message Type based on the error or status message desired, determine the SPI which is associated with this notification, determine if additional Notification Data is to be included, construct a Notification Payload, and finally transmit the messages to the receiving entity.
    - When a Notification payload is received, the receiving entity (responder) must determine if the Informational Exchange has any protection applied to it by checking the Encryption Bit and Authentication Only Bit in the ISAKMP Header, determine if the Domain of Interpretation (DOI) is supported, determine if the protocol-ID is supported, determine if the SPI is valid, determine if the Notify Message Type is valid, and then process the Notification payload, including additional Notification Data, and take appropriate action according to local security policy.

ISAKMP

- **ISAKMP Payload Processing**
  - **Delete Payload Processing**
    - Used when a SA is compromised
    - When a Delete Payload is created, the transmitting entity (initiator) must determine the DOI for this Deletion, determine the Protocol-ID for this Deletion, determine the SPI size based on the Protocol-id field, determine the number of SPIs to be deleted for this protocol, determine the SPI(s) which is (are) associated with this deletion, construct a Delete payload, and then transmit the message to the receiving entity.
    - When a Delete payload is received, the receiving entity (responder) must do as follows: Since the Information Exchange is protected by authentication for an Auth-Only SA and encryption for other exchange, the message must have these security services applied using the ISAKMP SA. Any errors that occur during the Security Service processing will be evident when checking information in the Delete payload.

Summary

- **OAKELY**
- **ISAKMP**
- **ISAKMP Header**
- **Generic Payload Header**
- **Payload Type:** SA, Proposal, Transform , IKE , Identification ,Certificate Payload, Certificate Request Payload, Hash, Signature Payload, Nonce, Notification, Delete Payload, Vendor ID
- **ISAKMP Exchanges :** Base Exchange, Identity Protection Exchange, Authentication Only Exchange, Aggressive Exchange, Informational Exchange
- **ISAKMP Payload Processing**