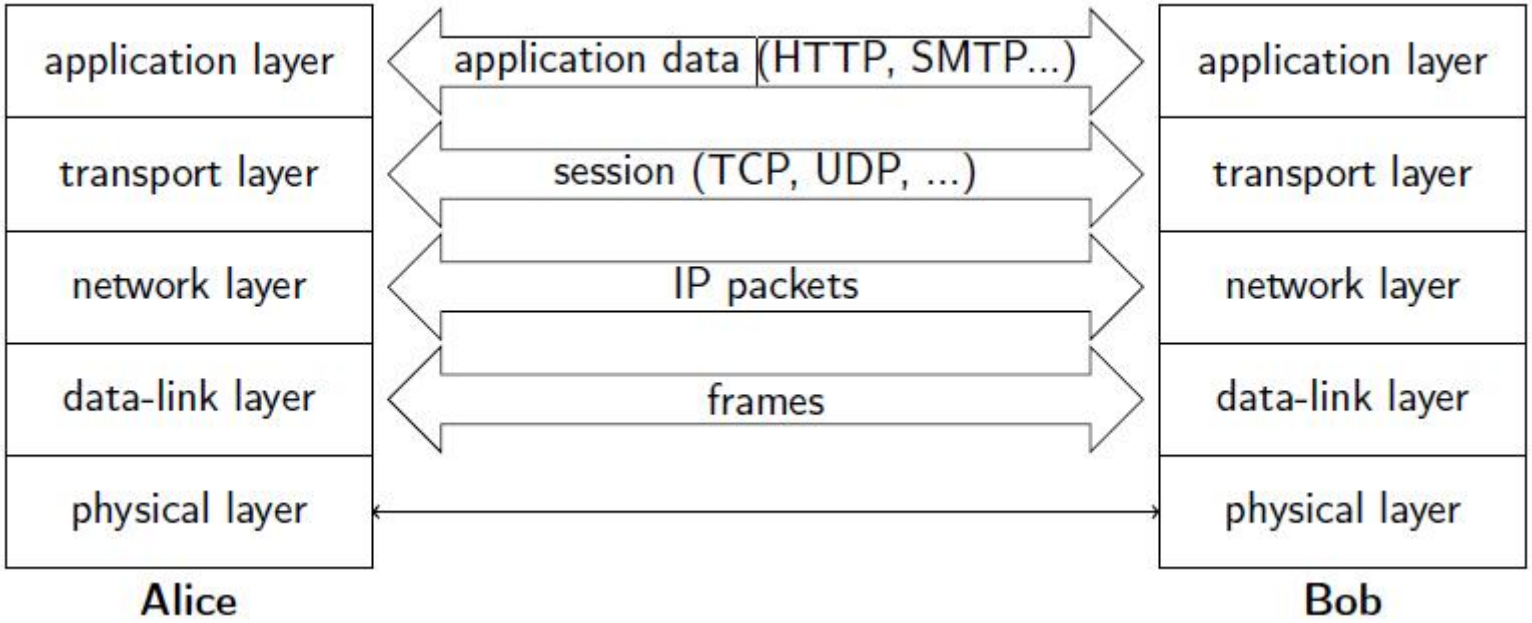
第4章 IPSec协议



- application layer security (SSH, S-MIME, PGP, ……)
- transport layer security (TLS/SSL, ……)
- network layer security (IPsec, ……)
- data-link layer security (WEP, WPA, WPA2, ……)

内容提要

- Motivation（IP协议的安全缺陷、虚拟专用网）
- IPSec概述（协议流程、SPD、SAD）
- 数据封装（IPSec：AH、IPSec：ESP）
- 安全参数协商（ISAKMP、IKE）

Motivation

Motivation

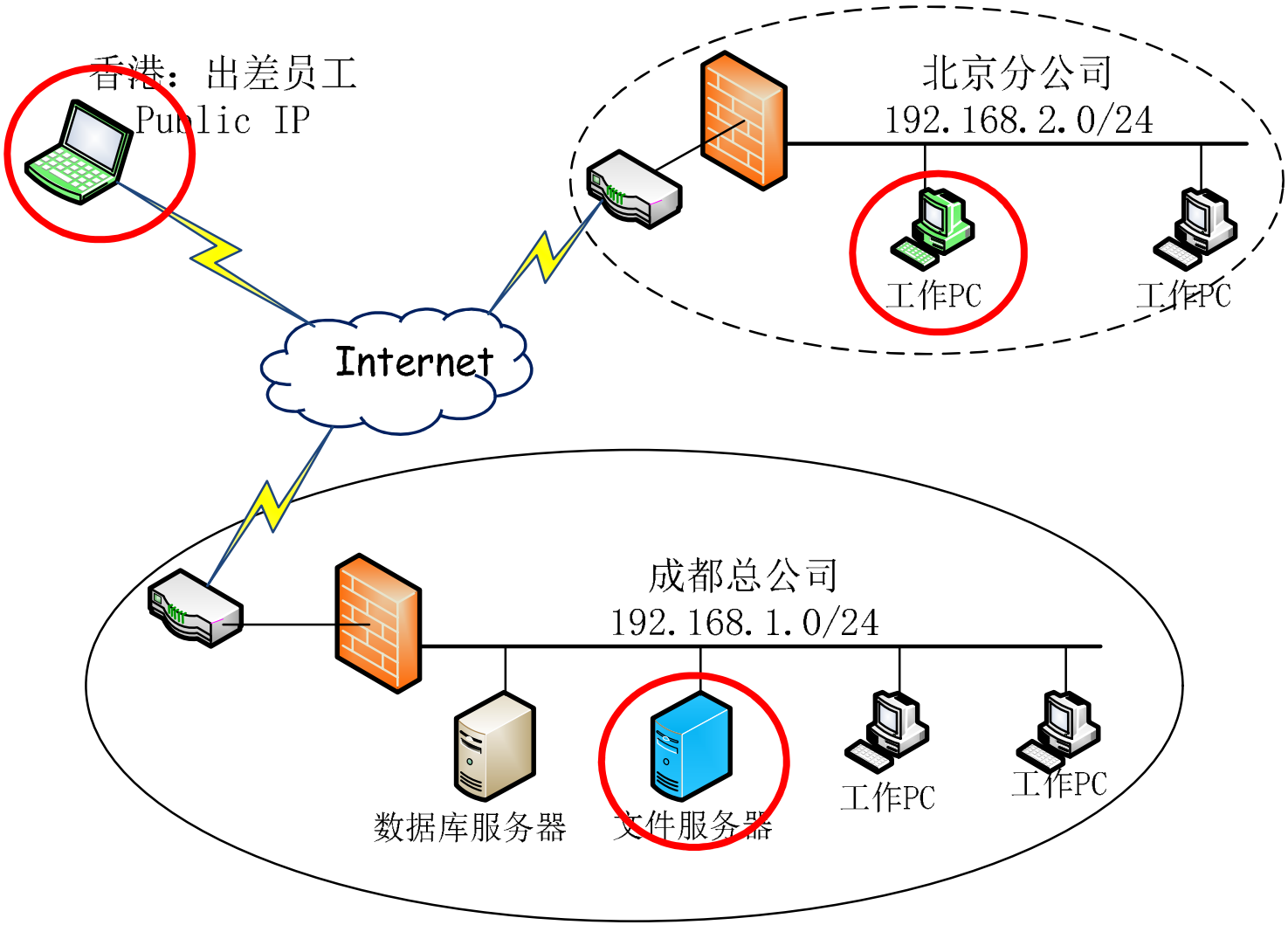
- IP协议的安全缺陷
- 虚拟专用网

IP协议的安全缺陷

- IP协议
  - Best Effort Delivery
- IP协议可能遭受欺骗攻击
  - 伪造源IP地址
- 缺乏保密性和数据完整性保护
  - 遭受被动攻击
  - 遭受主动攻击

克服这些缺陷需要相应的安全机制

虚拟专用网应用需求

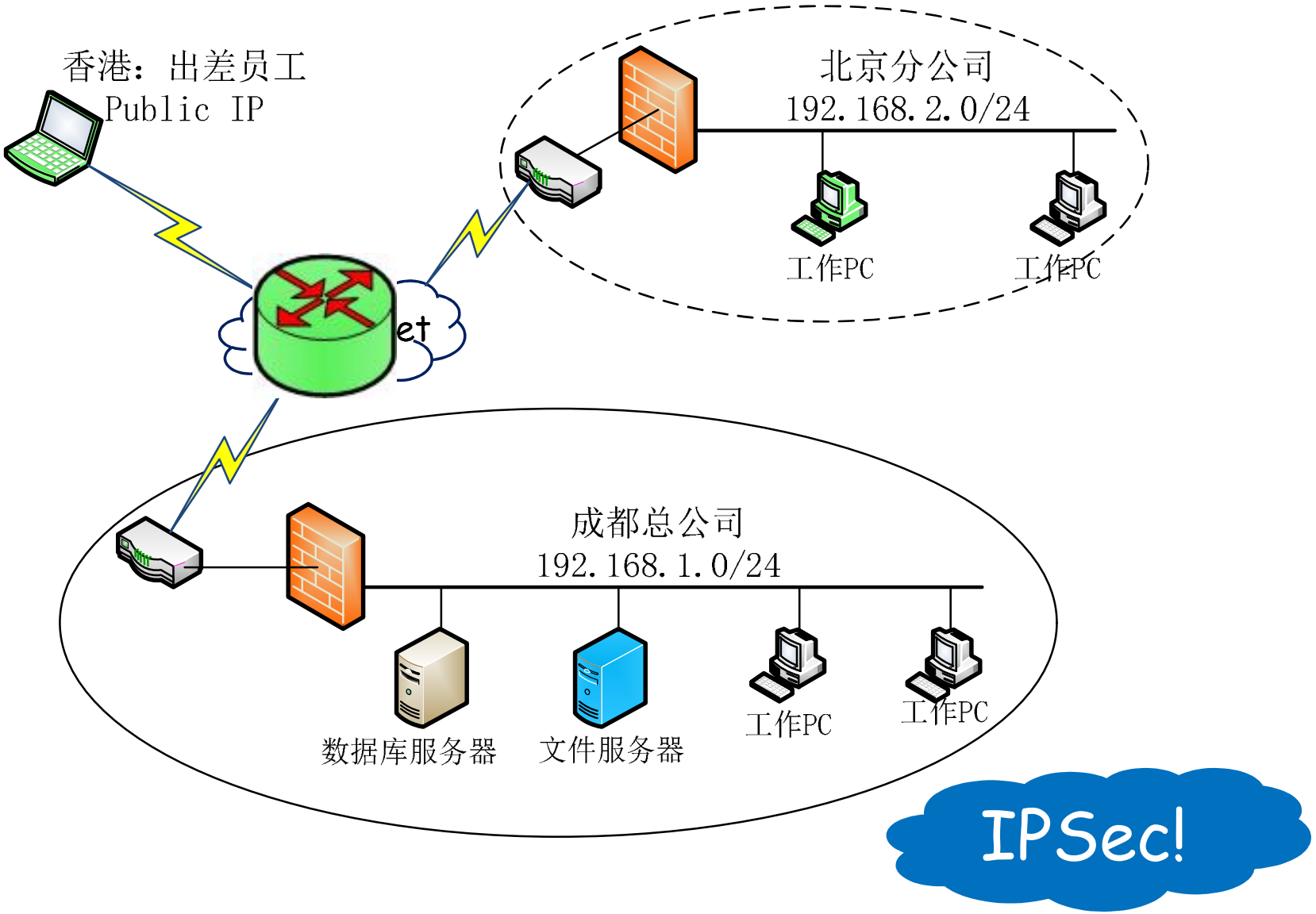


如何安全访问公司内部文件服务器?

解决方案1

- 通过设置NAT的端口转发
  - 外部的用户可以访问企业内部的服务器；
  - 但是这种方案**不具备安全性**，如果没有应用层安全措施（加密、消息认证码），传输的数据可以被攻击者窃听；
  - 由于开放端口，服务器也可能**成为攻击目标**。

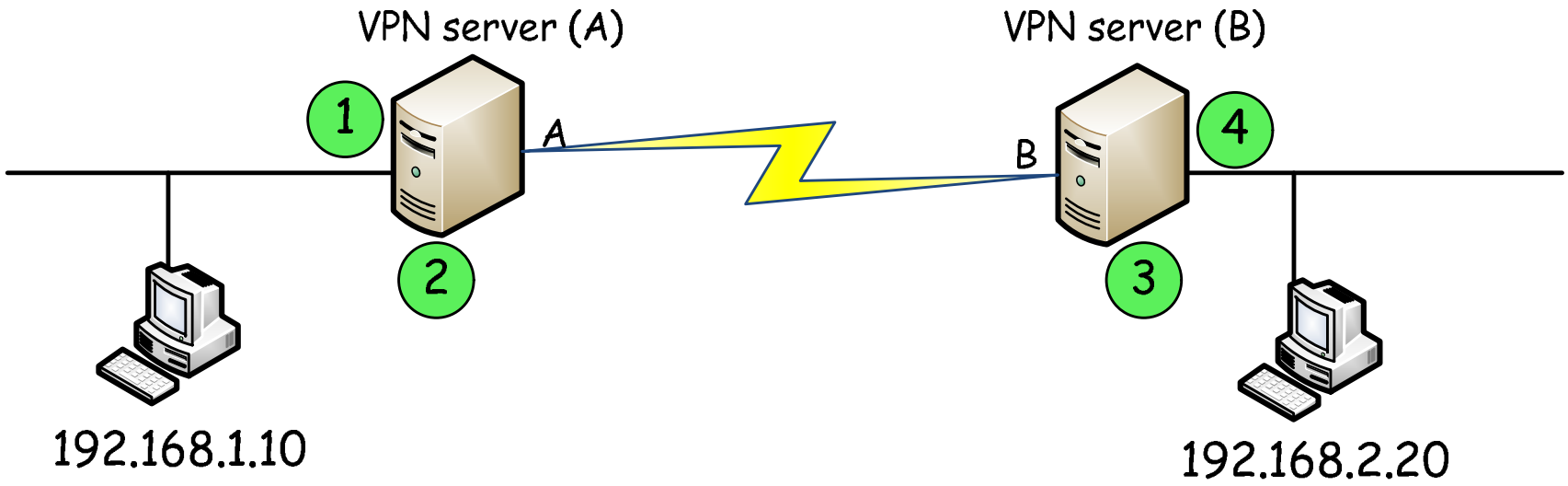
解决方案2


香港：出差员工
Public IP
北京分公司
192.168.2.0/24
工作PC
工作PC
et
成都总公司
192.168.1.0/24
数据库服务器
文件服务器
工作PC
工作PC
IPSec!

问题：

这里的源IP和目的IP都是私有的，不能通过互联网的路由器进行路由，那么又如何通过VPN来实现相互访问的呢？

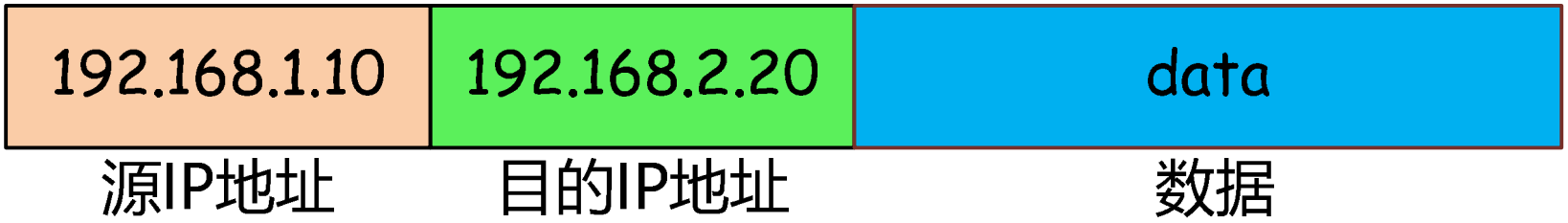
简单VPN示例（1）



- Server A和B为安全网关，双网卡
- Server A和B已经做好VPN配置

简单VPN示例（2）

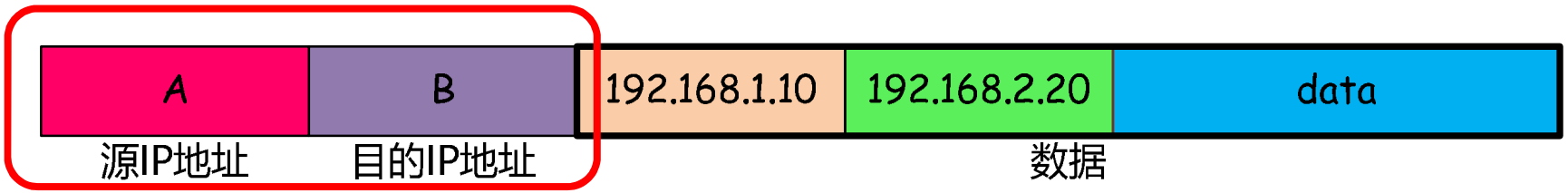
① 192.168.1.10主机生成要发送给192.168.2.20的数据包



② 根据192.168.1.10的路由表，该数据包会发送给VPN Server A

简单VPN示例（3）

③ VPN Server A对数据包进行重新封装，然后发送



④ VPN Server B收到数据包后，取出payload，然后将该数据包发送到192.168.2.0/24网段

简单VPN示例（4）

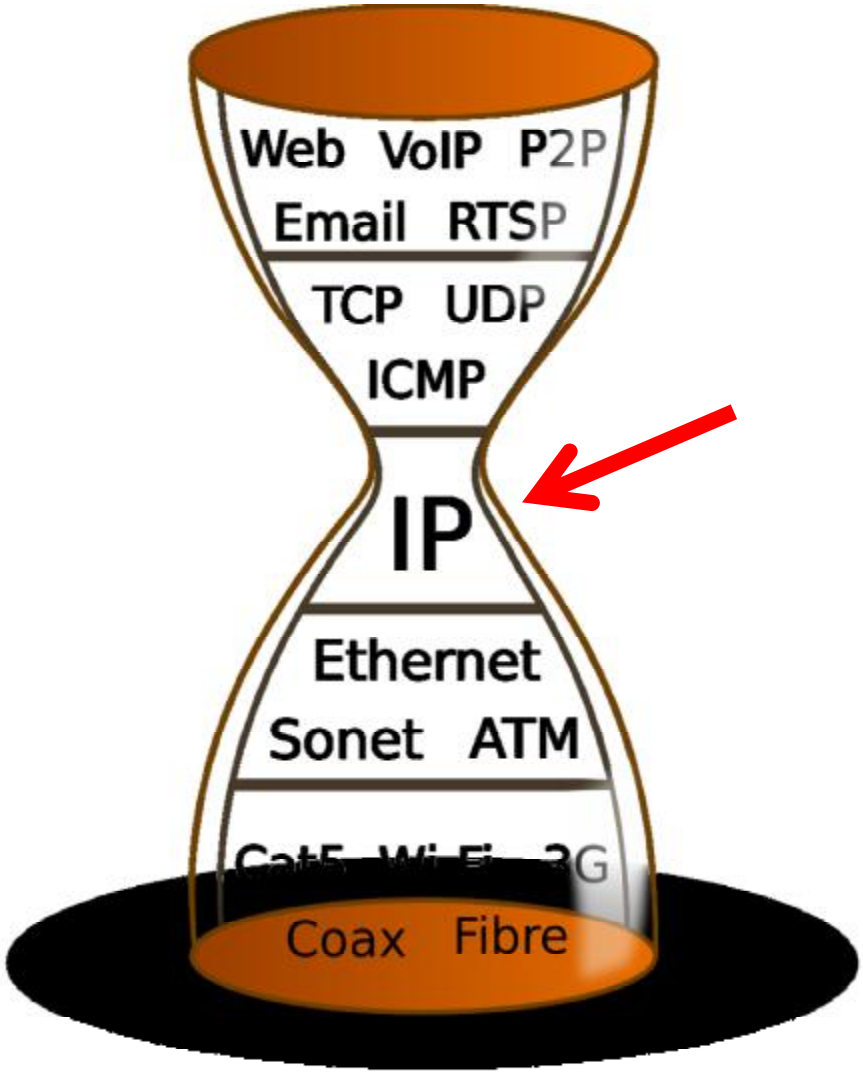
⑤ 进一步，为保证安全性，VPN Server A对数据包进行安全处理（如加密）

- 简单VPN相当于在A和B之间构建了一个加密隧道；进一步抽象，就好像A和B合并成了一个逻辑上的路由器。

VPN应用场景总结

- 公司有多个分部
  - 分布在世界各地
  - 各分部具有内部网络
  - 各分部接入互联网
  - 各分部之间需要互访
- 在外出差的员工需要访问公司内部资源

在IP层实现安全的好处



IPSec概述

IPSec安全服务

- 保密性（confidentiality）
- 数据完整性（data integrity）
- 访问控制（access control）
- 数据源认证（data origin authentication）

IPSec: 根据安全策略对IP数据报进行安全处理

安全策略：针对安全需求给出的一系列解决方案，它决定了对什么样的通信实施安全保护以及何种安全保护。

安全处理：加密、消息认证码、重新封装

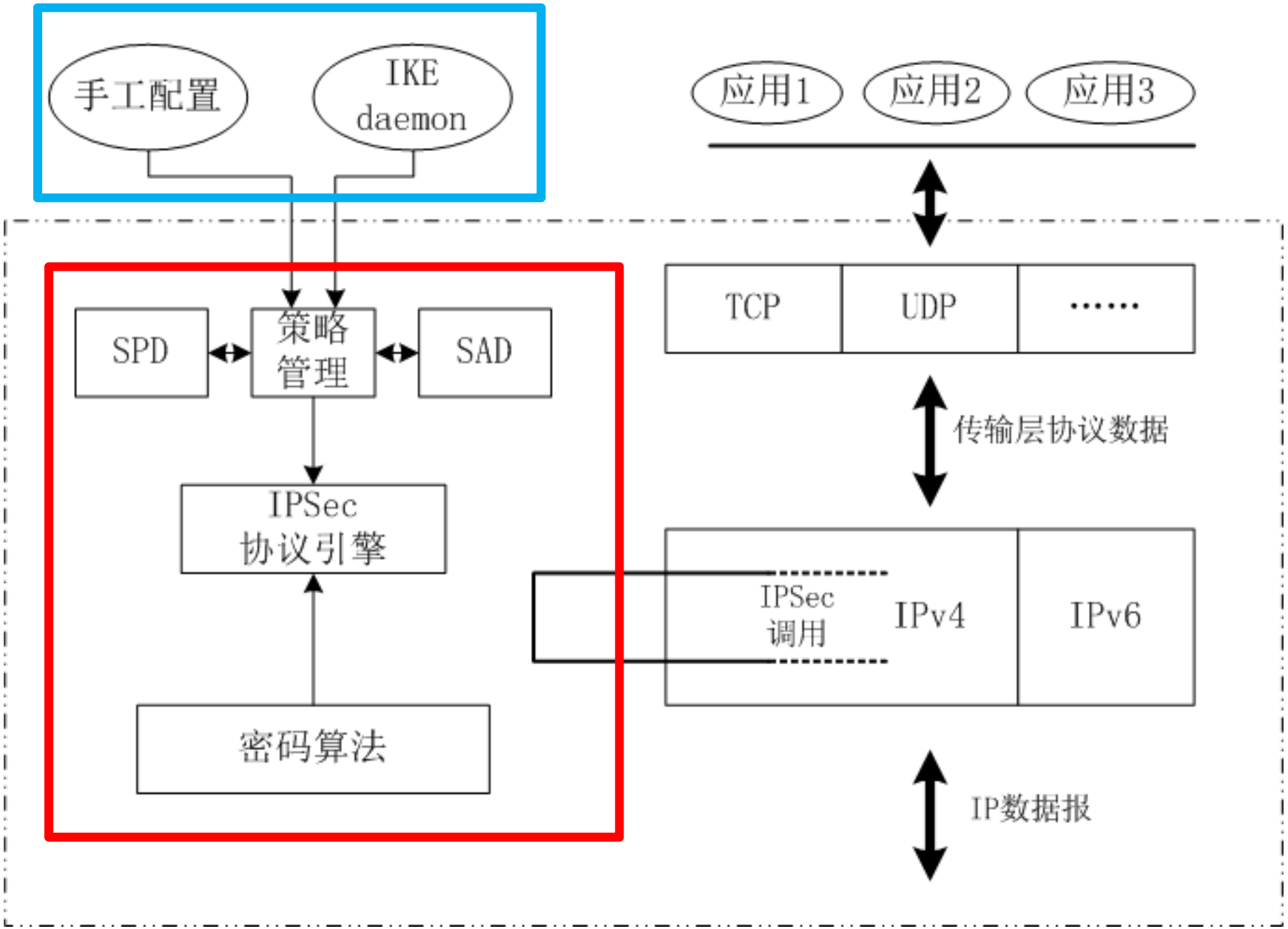
IPSec的处理过程分为协商和数据交互两个阶段

协商阶段：通信双方互相认证对方身份，并根据安全策略协商使用的加密、认证算法，生成共享的会话密钥。

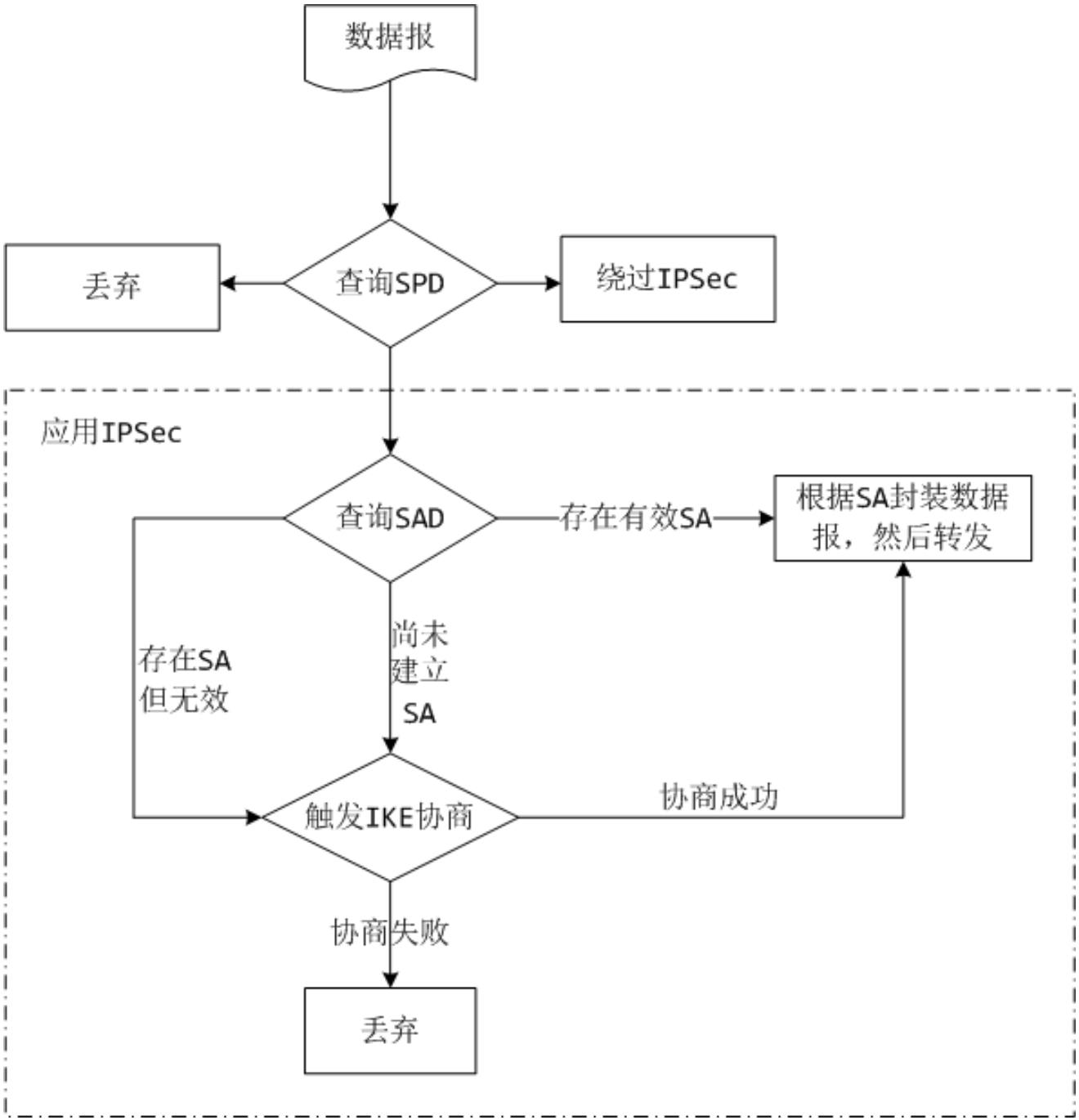
数据交互阶段：通信双方利用协商好的算法和密钥对数据进行安全处理。

IPSec

- 协商阶段
  - 互联网密钥交换协议（IKE）
- 数据交互阶段
  - 认证首部（AH）
  - 封装安全载荷（ESP）


手工配置
IKE
daemon
应用1
应用2
应用3
SPD
策略
管理
SAD
TCP
UDP
……
传输层协议数据
IPSec
协议引擎
IPSec
调用
IPv4
IPv6
密码算法
IP数据报

IPSec 外出数据处理流程



外出数据：应用IPSec，查询SAD

1. 存在有效SA
   - 根据对应参数，将数据报封装（加密、验证、添加IPSec头、IP头），然后送出
2. 尚未建立SA
   - 触发IKE协商，协商成功则走第1步，否则丢弃数据报
3. 存在SA但无效
   - 请求协商新的SA，协商成功则走第1步，否则丢弃数据报

进入处理

- 收到一个数据报后，首先查询SAD。如果存在有效SA，则查询为该数据报提供的安全保护是否与安全策略要求的相符。如果相符，则将还原后的数据报交给相应的高层协议模块处理或者转发；如果不相符，则丢弃该报文。
- 如果没有建立SA或者SA无效，则直接丢弃数据报，不会重新协商SA。

IPSec：

IPSec引擎根据**安全策略(SP)**和**安全关联(SA)**

调用密码算法对IP报文做**安全处理**

1. 安全策略（SP）
2. 安全关联（SA）
3. 安全处理→报文封装
4. 安全关联→IKE

IPSec：安全策略（SP）

安全策略（Security Policy）

- 安全策略是针对安全需求给出的一系列解决方案，**决定了对什么样的通信实施安全保护以及提供何种安全保护**
- 安全策略通常以安全策略库（Security Policy Database， SPD）的形式表现，其中每条记录是一条安全策略。
- SPD一般保存在一个策略服务器中，为域中的所有节点（主机和路由器）维护策略库。各节点可以将策略库复制到本地，也可以通过LDAP动态获取策略。

SPD

- 对于外出数据报，必须先检索SPD，确定对其应提供的安全服务。
- 对于进入数据报，IPSec引擎也检索SPD，判断为其提供的安全保护是否和策略规定的安全保护相符。
- 安全策略描述
  - 对通信特性的描述
  - 对保护方法的描述

SPD：通信特性描述

- 目的IP
- 源IP
- 名字，如DNS名、X.500区分名等
- 传输层协议，如果TCP、UDP
- 源、目的端口，可以是单个端口、端口列表或者通配端口
- 数据敏感等级

以上对通信特性的描述称为：选择符（selector）

SPD:对保护方法的描述

- 每一份数据报，三种处理方式：
  - 丢弃
  - 绕过
  - 应用IPSec
- 如何应用IPSec?
  - 安全关联（SA）

SPD简单示意

源IP	目的IP	执行协议	源端口	目的端口	工作模式
192.168.0.1	192.168.0.10	AH/ESP	Any	110	Transport
192.168.0.10	192.168.0.1	AH/ESP	110	Any	Transport
192.168.0.1	192.168.0.20	ESP	Any	1433	Transport
192.168.0.20	192.168.0.1	ESP	1433	Any	Transport

IPSec：安全关联（SA）

安全策略与安全关联的关系

- 安全关联（SA）用于实现安全策略，是安全策略的具体化和实例化，详细定义了对一个数据报的具体处理方式。
- 如果安全策略是应用IPSec进行保护，则必须指向一个SA或者SA束。

安全关联（SA）

- SA是两个IPSec实体（主机或者路由器）之间的一个单工“连接”，用于规定保护数据报安全的具体**安全协议**、**密码算法**、**密钥**以及**密钥的生存期**。
- SA是单向的，要么对数据报进行“进入（接收到的）”保护，要么对数据报进行“外出（发送出去的）”保护

安全关联

- 每个SA用一个三元组来标识
  - <SPI, 目的IP地址，安全协议>
- 其中：SPI, Security Parameter Index
  - 4字节的串
  - 协商SA时指定
  - IPSec报文中包含SPI
- 其中：目的IP地址
  - 单播、广播或者多播地址

安全关联字段（1）

- 目的IP地址
  - 可以为终端用户系统、防火墙或者路由器
- IPSec协议
  - AH
  - ESP
- 序号计数器
  - 32bit，用于产生AH或ESP头的序号
- 序号计数器溢出标志
  - 标识序号计数器是否溢出。如溢出则产生一个审计事件，并禁用该SA保护数据

安全关联字段（2）

- 抗重放窗口
  - 32bit，用于确定进入的AH或者ESP包是否为重放。
- 密码算法及密钥
  - 消息验证码计算算法及密钥
  - 加密算法及密钥
  - 初始化向量IV
- 安全关联的生存期
  - 一个时间间隔，超过这一间隔后，应建立一个新的SA或者终止通信。生存期可以用时间或者当前SA处理过的字节数为单位来计数

安全关联字段（3）

- IPSec协议模式
  - 传输模式（transportation）
    - 提供对高层协议数据的保护
  - 隧道模式（tunnel）
    - 提供对整个IP报文的保护

安全关联简单示例

SPI	目的IP	执行协议	ESP验证算法	ESP验证算法密钥	ESP加密算法	ESP加密算法密钥	工作模式
0X201	192.168.0.10	ESP	Na	Na	3DES-CBC	iiiiiiiiiii	Transport
0X301	192.168.0.1	ESP	Na	Na	3DES-CBC	jjjjjjjjjj	Transport
0X401	192.168.0.20	ESP	HMAC-SHA1	lllllllllll	3DES-CBC	rrrrrrrr	Transport
0X501	192.168.0.1	ESP	HMAC-MD5	ttttttttttt	3DES-CBC	ppppppp	Transport

安全关联管理

- 安全关联管理的任务
  - 创建SA
  - 删除SA
- 手工方式
  - 由管理员根据安全策略来维护，容易出错
- 自动方式
  - IKE
  - 动态
  - 适用于规模较大的情况

IPSec的封装

IPSec封装: AH
——Authentication Header

AH提供的安全服务

- 数据完整性（Integrity）
  - 消息验证码
- 数据源发认证（Data Origin authentication）
  - 消息验证码
- 抗重放攻击（Anti-replay attack）
  - 序列号

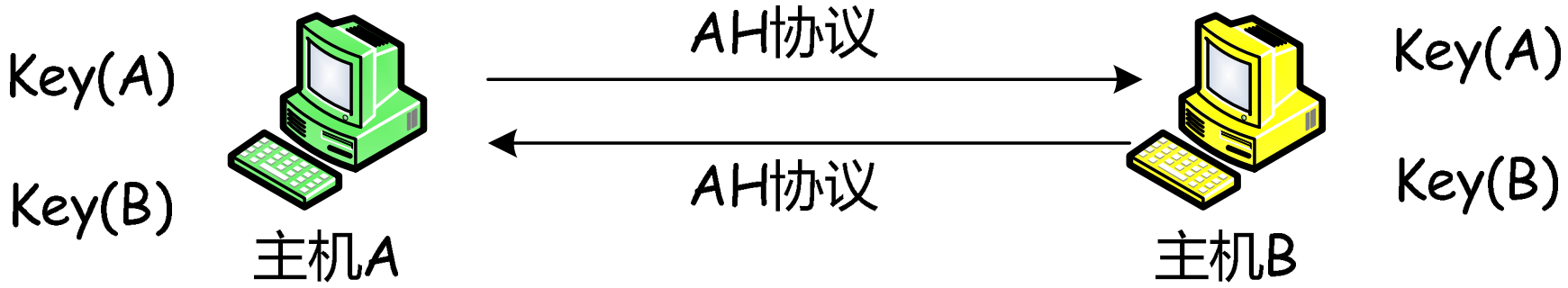
AH的特点

- 不对传输的数据提供保密性
- 密码学运算开销小
- 两种工作模式
  - 传输模式（transport mode）
  - 隧道模式（tunnel mode）

IPSec: AH
——Transport mode

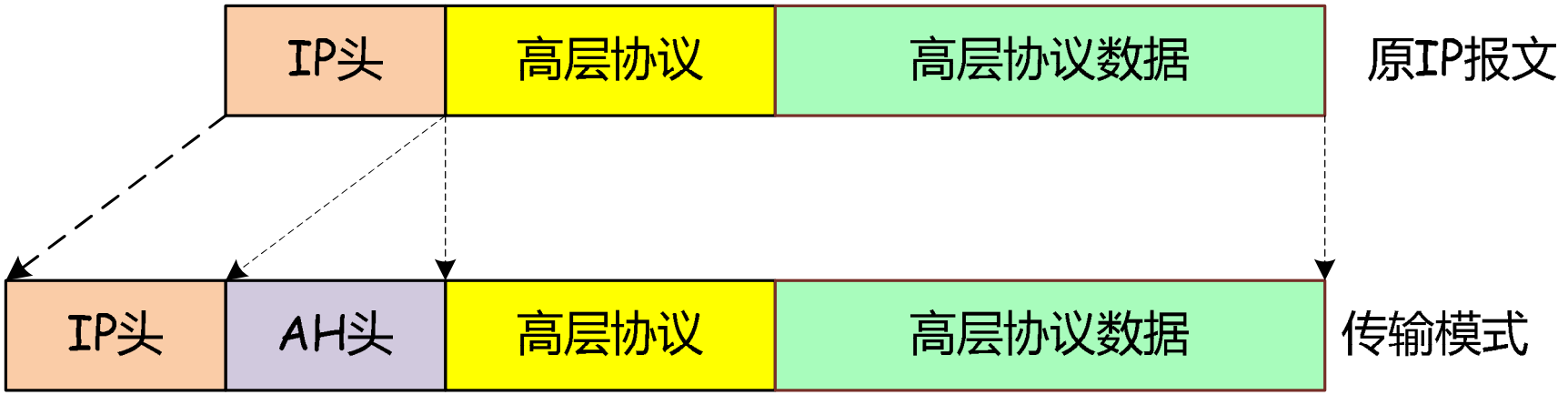
传输模式AH

- 如果要在两台主机之间以AH协议保护传输的数据，需要先在两台主机设置密钥，假设主机A的密钥为Key(A)，主机B的密钥为Key(B)。因为AH协议使用对称加密，因此通信双方还需要对方的密钥。



传输模式AH：发送端报文封装

- 主机A的IPSec引擎会对要发送的数据包进行处理，在原有的IP包头和高层协议（如TCP）头部之间插入一个AH头部。AH头部包含了原来IP报文的**完整性校验值**（ICV），封装好之后，发送给主机B。



传输模式AH：接收端校验

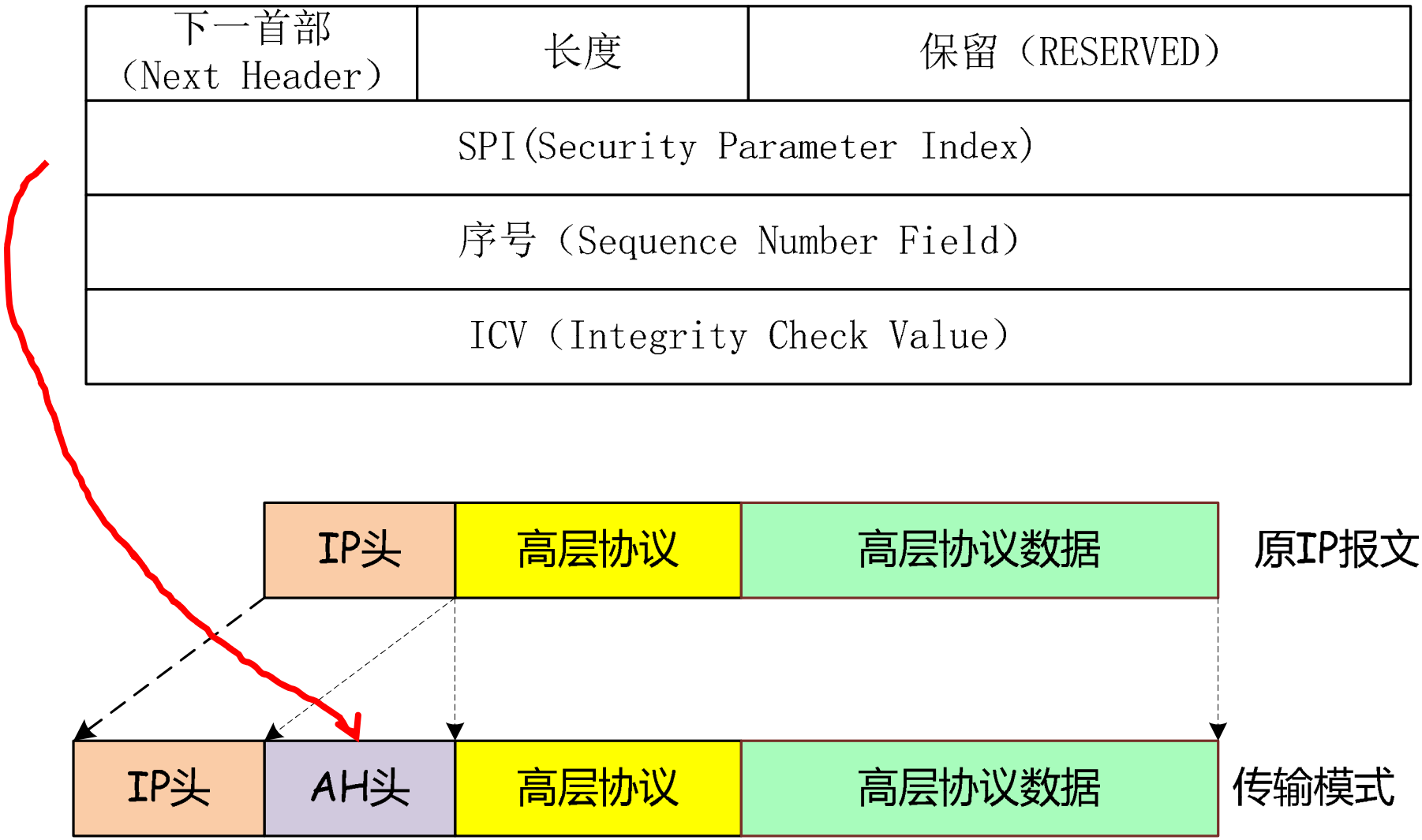
- 主机B收到上述报文后，根据AH头部的信息，对数据包进行验证，以确定是否被篡改。
- 验证过程为：
  - 主机B收到数据包之后，根据对应SA的信息生成认证数据，并与认证头部中的认证数据进行比较
  - 如果匹配，则验证成功
  - 如果不匹配，则认为数据包无效，丢弃该数据包，并在audit log里面记录这个事件。

Question?

1、具体报文格式?

2、如何计算ICV（integrity check value）？

传输模式AH：头部格式


下一首部
（Next Header）
长度
保留（RESERVED）
SPI(Security Parameter Index)
序号（Sequence Number Field）
ICV（Integrity Check Value）
IP头
高层协议
高层协议数据
原IP报文
IP头
AH头
高层协议
高层协议数据
传输模式

传输模式AH：头部格式

- next hdr：8-bit字段，指明认证头部后面的一个首部对应的协议类型。
- 长度：8-bit字段，指明认证头部的长度，以4个字节为计数单位，再减去2。
- Reserved：16-bit字段，保留为将来用，设置为0。
- SPI：32-bit字段，IKE协商SA时指定的安全参数索引，<SPI，目的IP地址，安全协议（这里是AH）>三元组用于唯一地标识该数据包的SA。

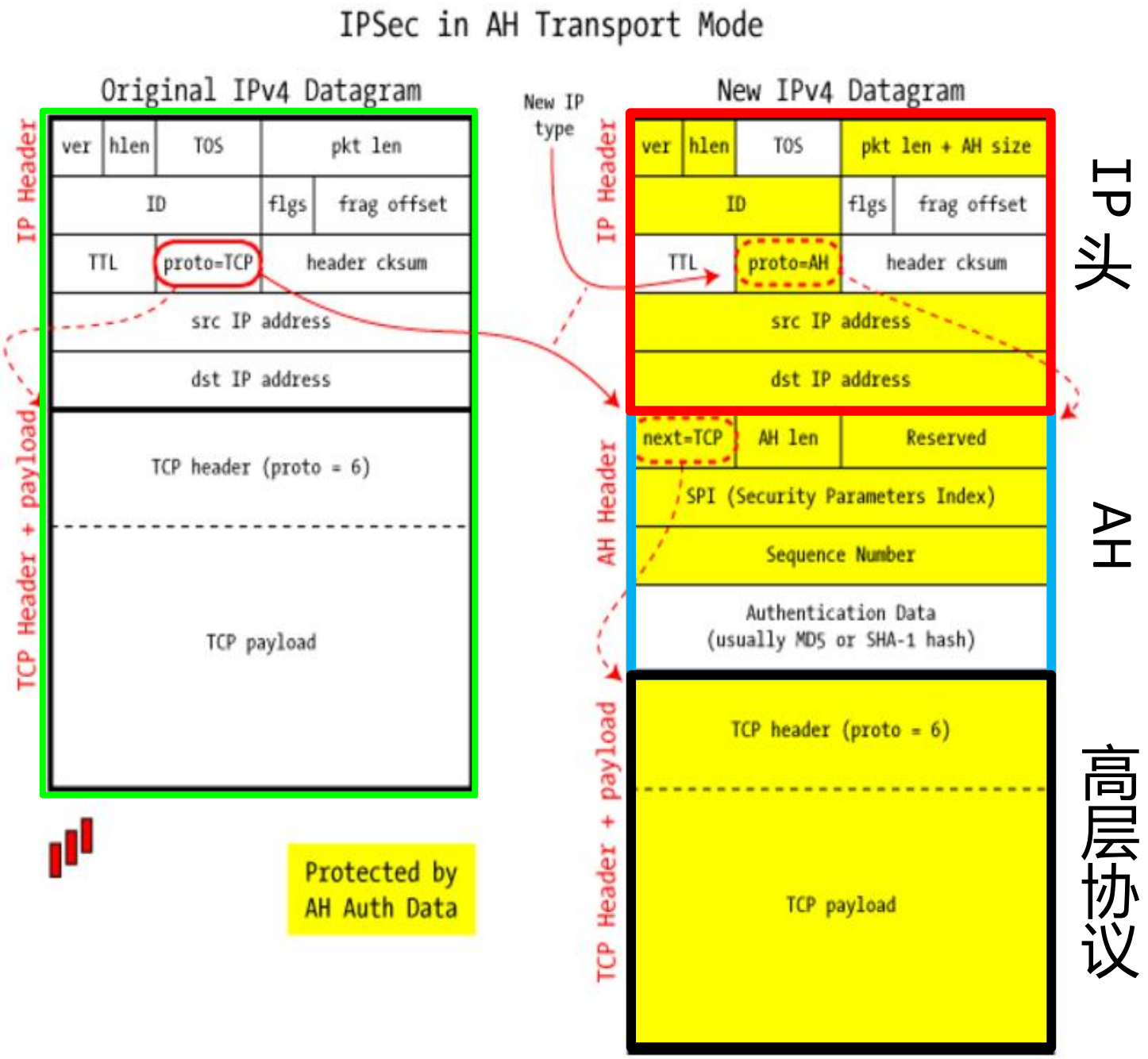
传输模式AH：头部格式（续）

- Sequence Number：无符号32-bit、单调递增计数器值，用于防止重放攻击。即使接收方对某个具体的SA不选择启动抗重放攻击，该字段也是必须的。
- ICV：变长字段，必须是32-bit的整数倍，用于存储完整性校验值，用于进行**数据源发认证**和**完整性校验**。

传输模式AH：ICV计算数据

- IP头部中，在传输过程中不变的字段或者是在到达接收端其值可预测的字段
- AH头部，包括：Next Header，Payload Len，Reserved，SPI，Sequence Number，和认证数据（在计算ICV值的时候置为0），及显式的填充字节（如果有的话）。
- 上层协议数据，这些数据被认为在传输过程中是不变的。

AH 传输模式的报文封装



AH保护数据：**发送**处理过程

1. 查询SA，获取安全参数；
2. 生成序列号；
3. 计算认证数据；
4. 构造IPSec报文并发送。

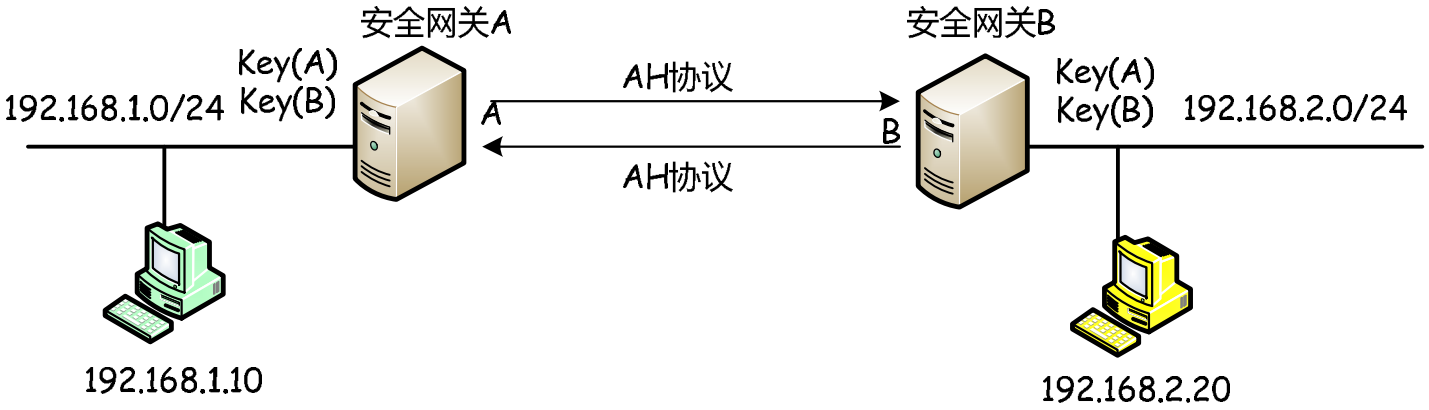
AH保护数据：**接收**处理过程

1. 根据<目标IP地址，安全协议，SPI>在SAD中查找对应的SA，如果没有找到，则丢弃该报文；
2. 使用滑动窗口机制验证序列号，防止重放攻击；
3. 验证认证数据，若通过验证，则还原数据并递交给相应的协议模块或者转发，否则丢弃。

IPSec封装：AH
——Tunnel mode

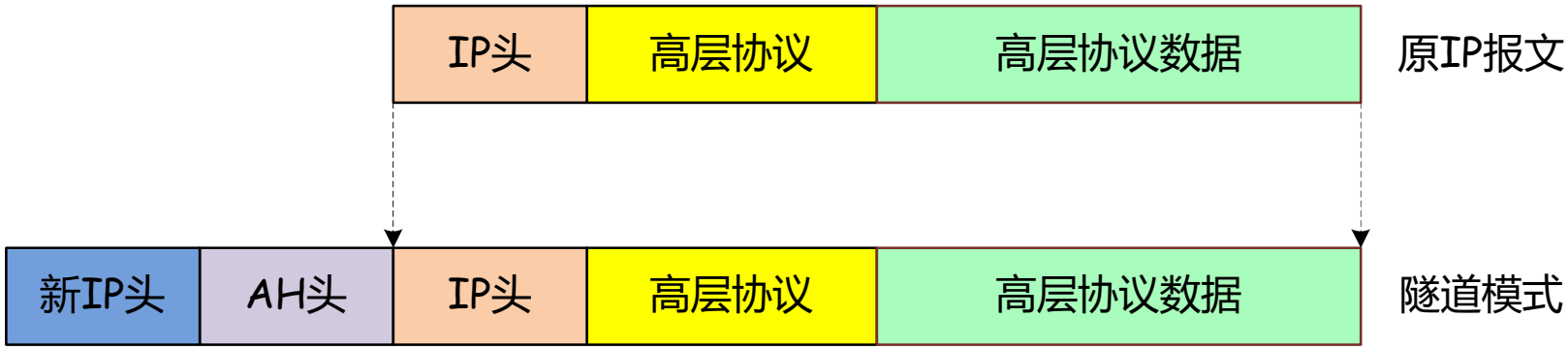
隧道模式AH

- 安全网关A用来保护192.168.1.0/24网段，该网段所发出的外出包，都会经过主机A。同样地，安全网关B用来保护192.168.2.0/24网段，该网段所有要出去的包均经过主机B转发。



AH隧道模式：报文封装

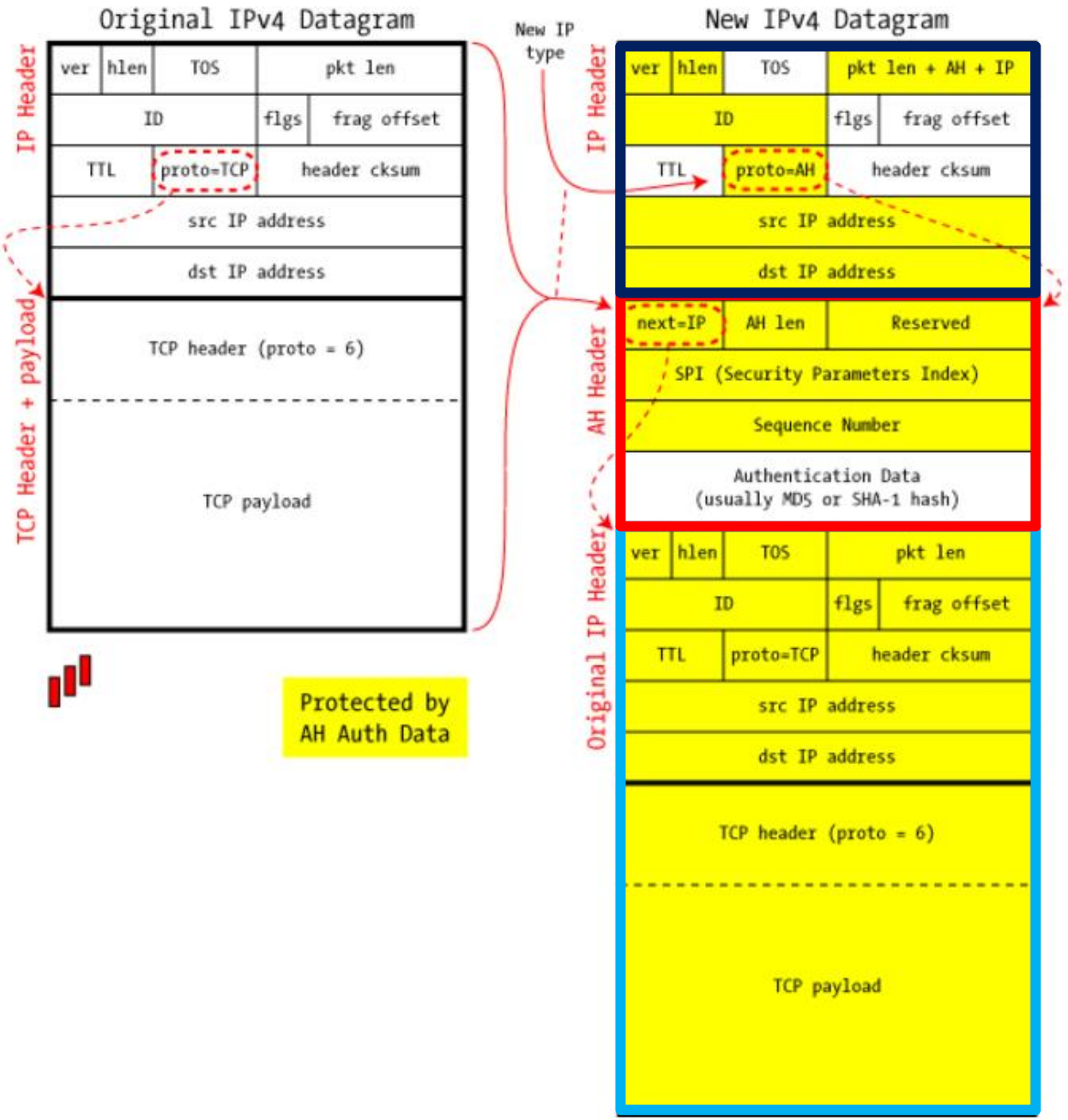
- 当192.168.1.10发送数据包给192.168.2.20时，该数据包首先到达主机A。主机A对该报文进行重新封装



AH隧道模式：完整性验证

- 当主机B收到报文后，主机B会根据AH头部信息进行验证，该过程与传输模式相同。如果验证通过，主机B将恢复原来的IP报文（即：去掉新的IP头部和AH头部），然后根据主机B的路由表发送出去。如果验证失败，则丢弃该报文。

AH 隧道模式的报文封装

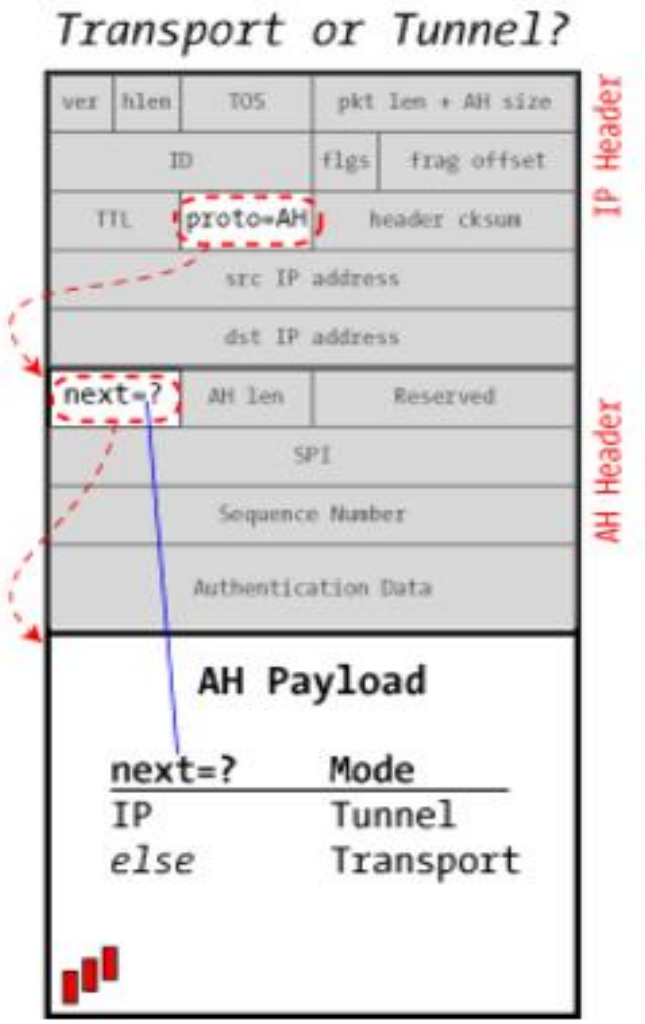


Question?

1、如何判断是传输模式还是隧道模式?

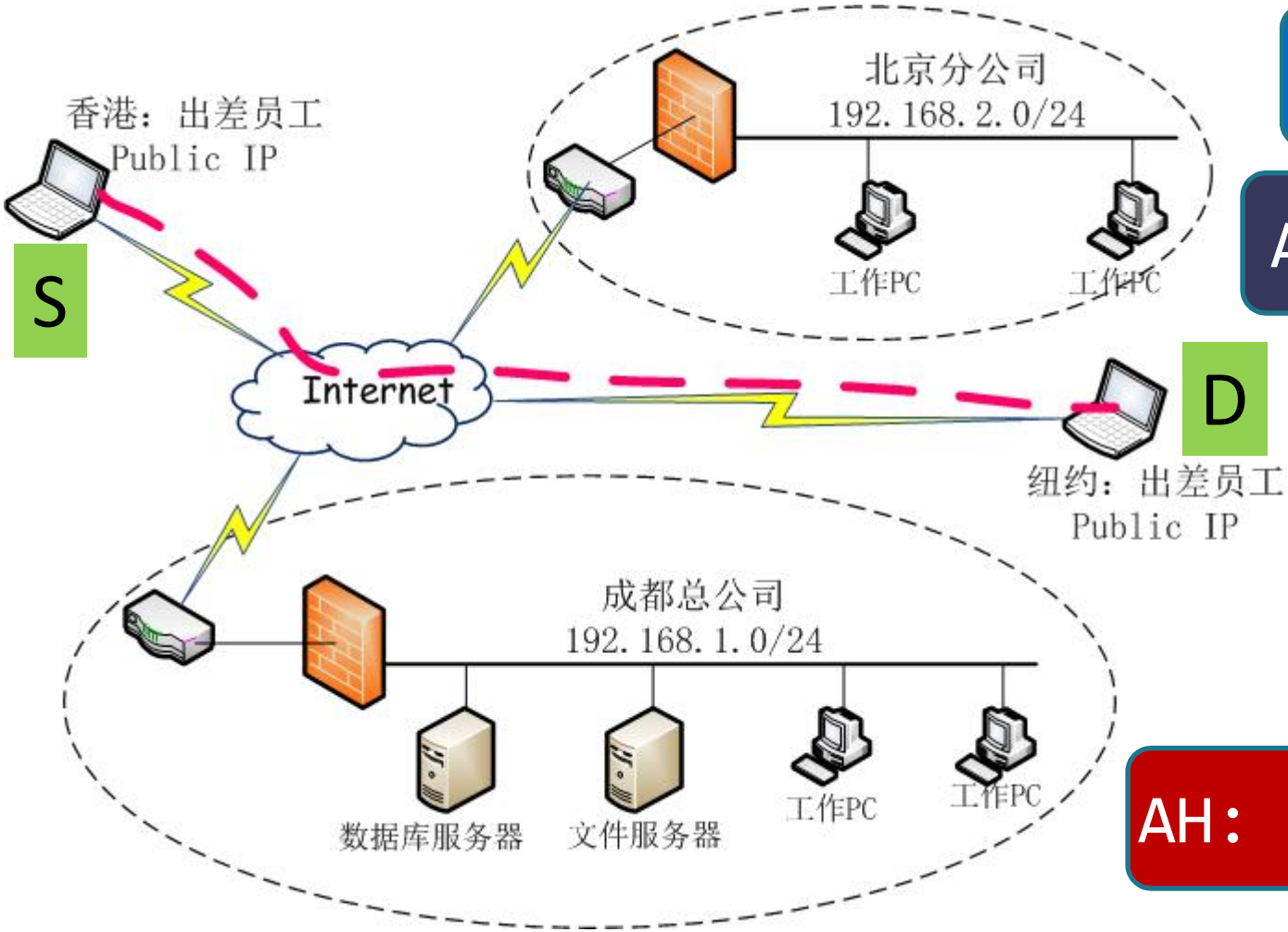
如何判断是传输模式还是隧道模式?

- 判断一个报文是隧道模式还是传输模式的关键在AH的next字段。如果next=IP，则表明AH后面是一个IP报文，那么就是隧道模式，把整个IP报文传送出去。其他情况则是传输模式。



2、如何应用AH?

应用场景（1）



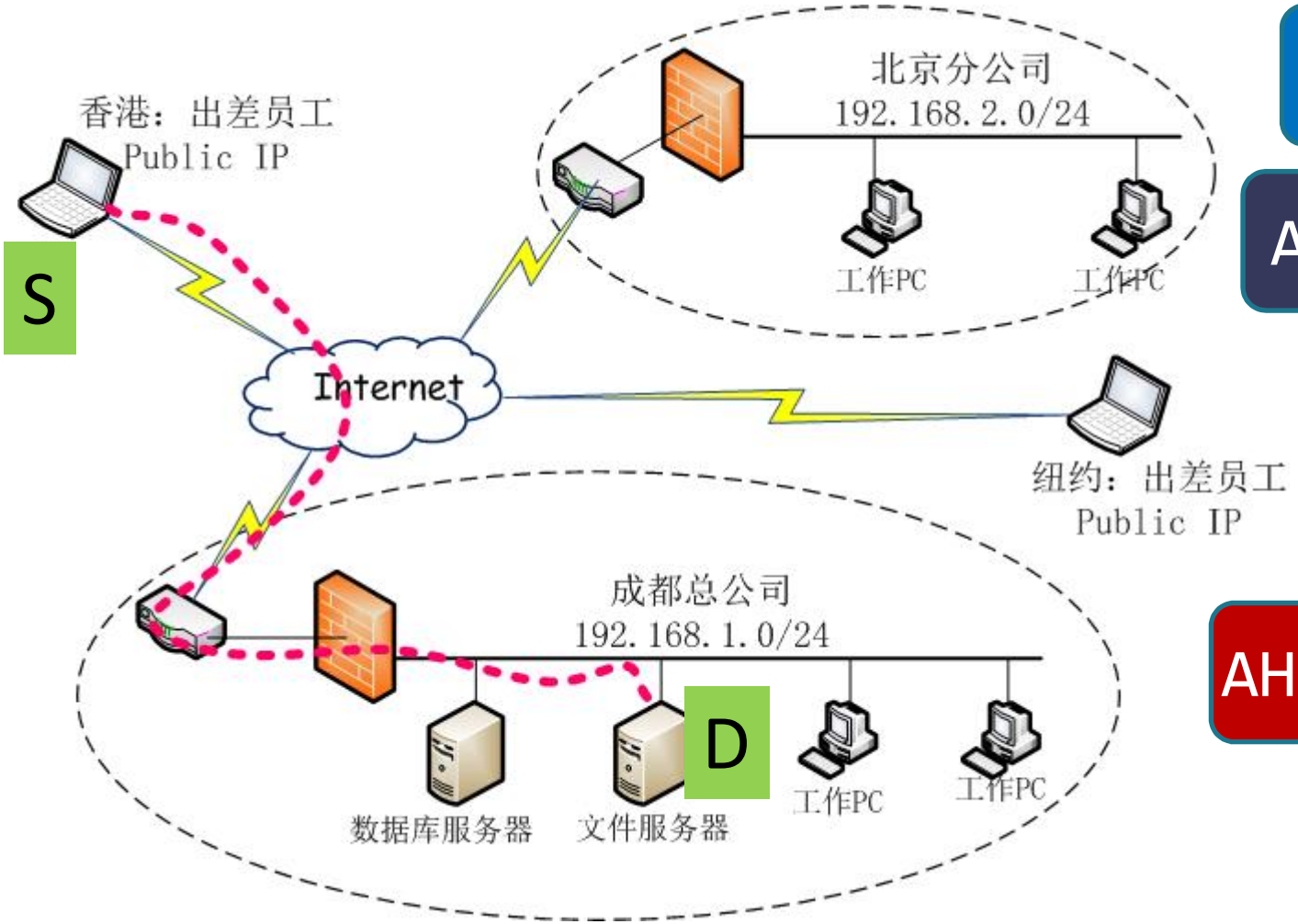
AH： ?模式

AH：传输模式

AH： 哪里部署?

AH： 安全的起点与终点?

应用场景（2）



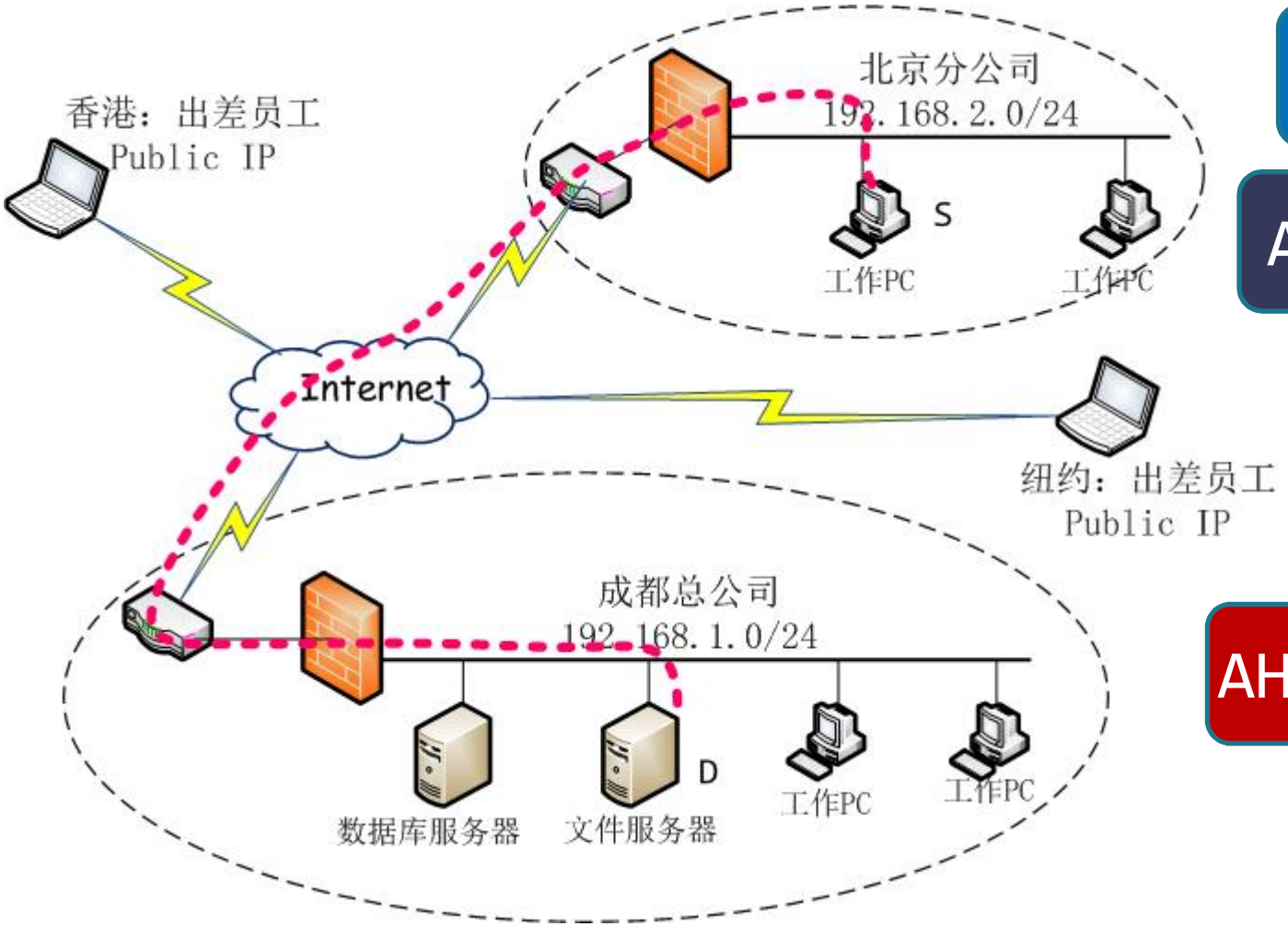
AH： ?模式

AH： 隧道模式

AH： 哪里部署?

AH： 安全的起点与终点?

应用场景（3）



AH： ?模式

AH： 隧道模式

AH： 哪里部署?

AH： 安全的起点与终点?

IPSec: ESP
(Encapsulating Security Payload)

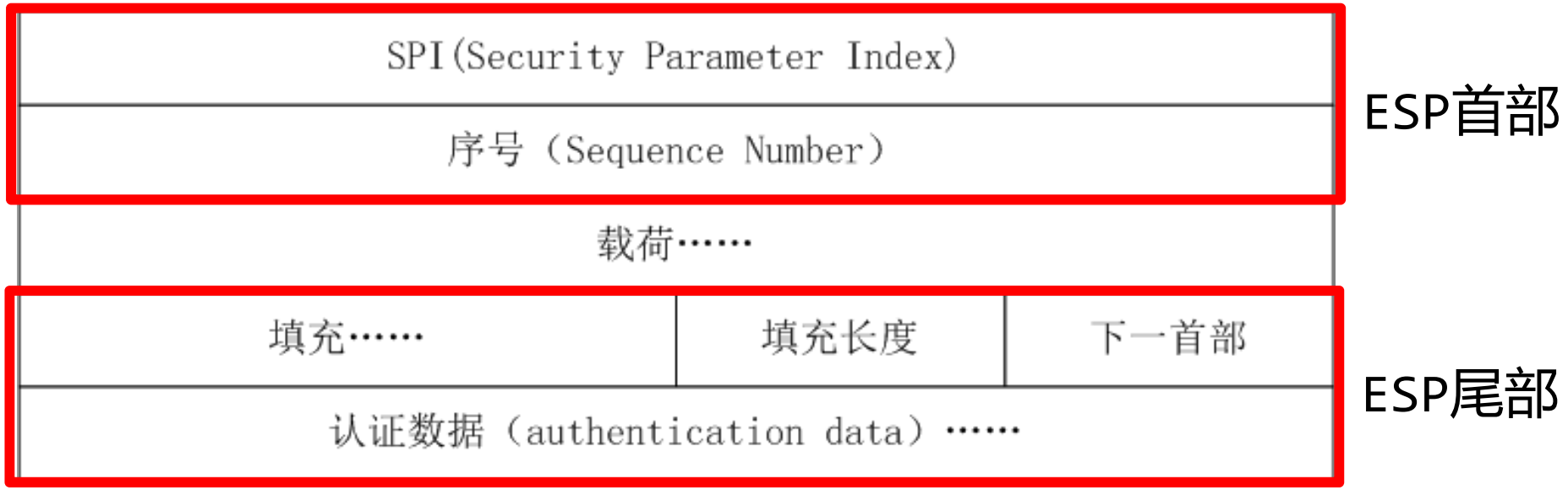
ESP：安全服务

- 保密性
- 数据完整性
- 数据源发认证
- 抗重放攻击

对比AH的安全服务：

- 数据完整性
- 数据源发认证
- 抗重放攻击

ESP报文格式


SPI(Security Parameter Index)
序号（Sequence Number）
载荷……
填充……
填充长度
下一首部
认证数据（authentication data）……
ESP首部
ESP尾部

ESP字段说明（1）

- SPI：协商SA时确定的安全参数索引
- 序号：报文编号，用于防止重放攻击
- 载荷：加密数据，与应用模式相关
- 填充：
  - 如果加密算法采用分组加密，则在明文数据不是分组长度的整倍数的情况下，需要填充
  - 不论采用何种加密算法，需要确保密文在一个4-byte的边界结束，如果不满足，则需要进行填充；目的是确保认证数据在一个4-byte边界对齐。

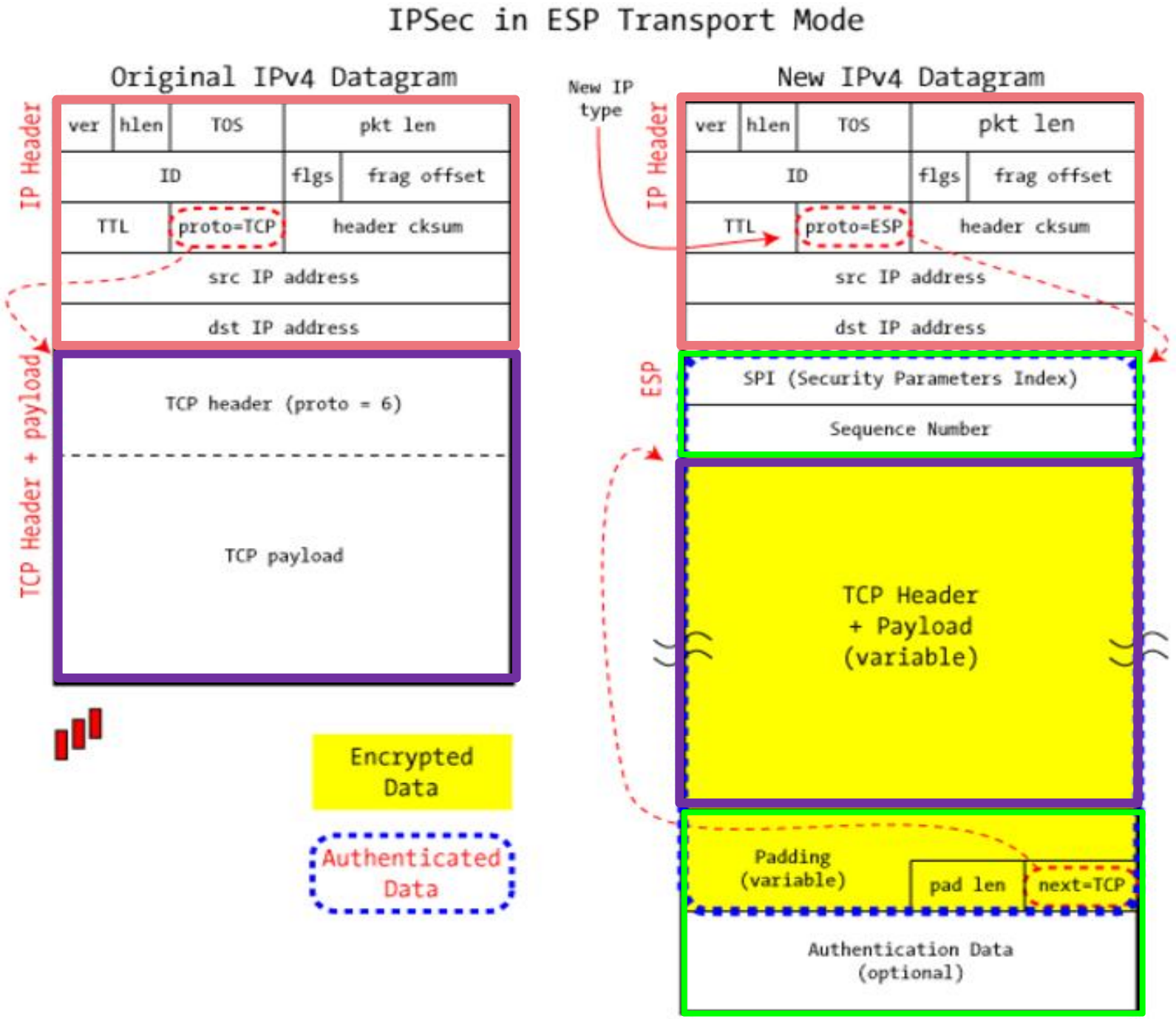
ESP字段说明（2）

- 认证数据
  - 变长字段
  - 可选字段
  - Integrity Check Value (ICV)
    - 输入为ESP packet (不包括 authentication data字段)
  - 认证算法必须说明ICV的长度、比较规则和验证的处理步骤

IPSec: ESP
Transport mode

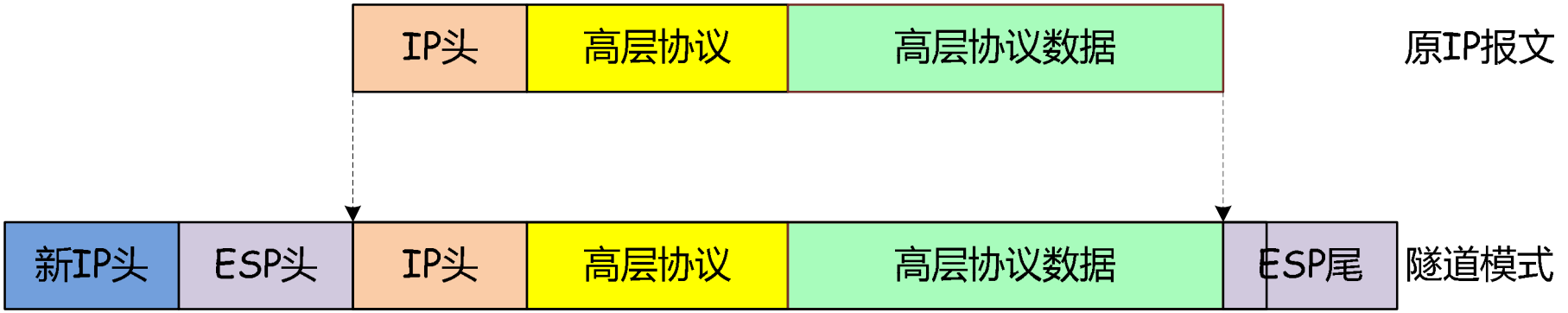
ESP传输模式报文

ESP：传输模式的报文封装

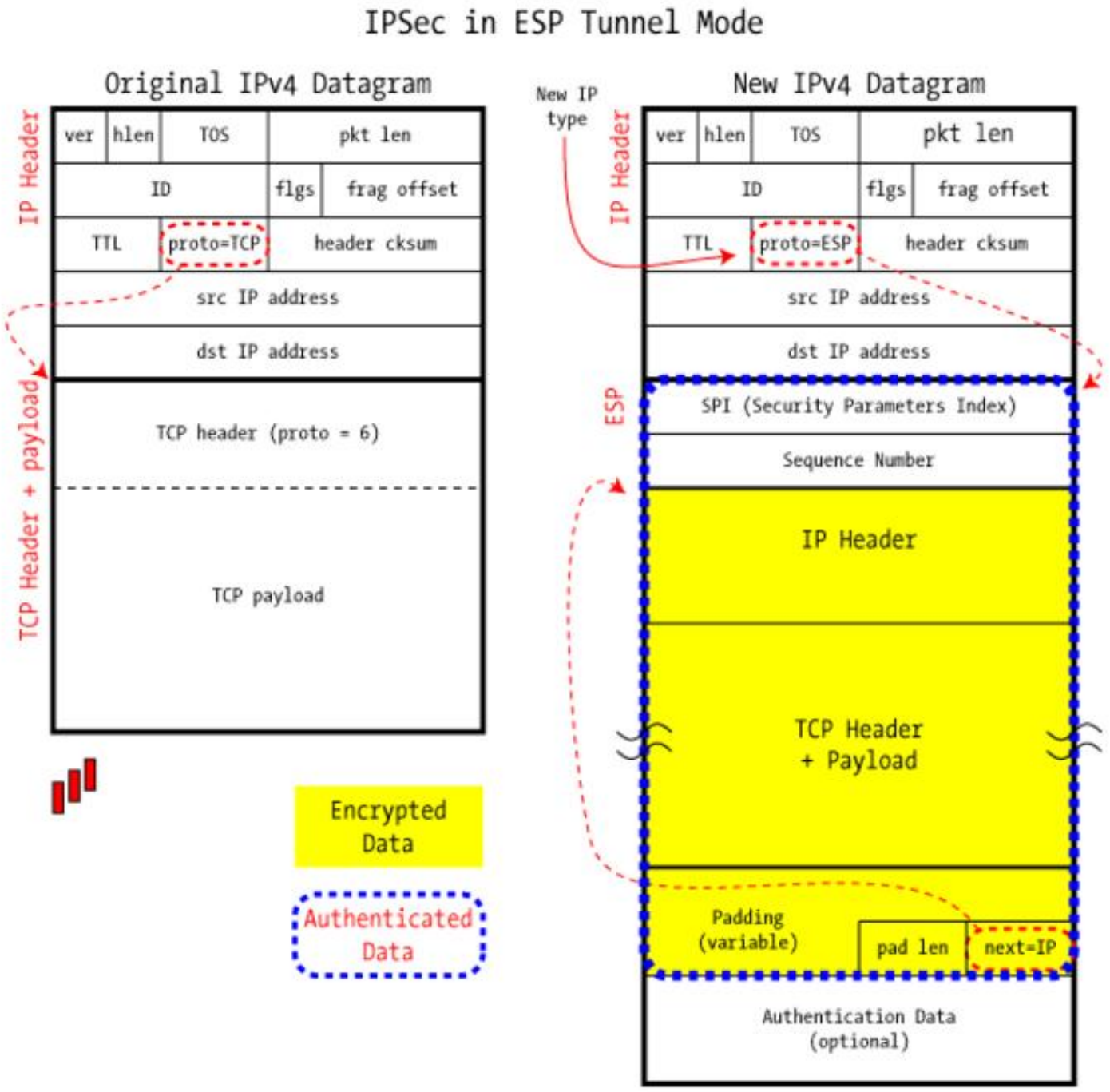


IPSec: ESP
Tunnel mode

ESP隧道模式报文



ESP：隧道模式的报文封装



ESP：**发送**数据处理过程

1. 查询SAD，获取安全参数
2. 报文加密
3. 生成序号
4. 计算认证数据
5. 构造IPSec报文并发送

ESP：**接收**数据处理过程

1. 根据<目的IP地址，安全协议，SPI>三元组查询SAD，若找不到有效SA，则丢弃报文
2. 使用滑动窗口机制验证序号，防止重放攻击
3. 验证认证数据，如果验证失败，则丢弃该报文
4. 解密，并将还原后的报文递交给相应的协议模块，解密失败则丢弃该报文

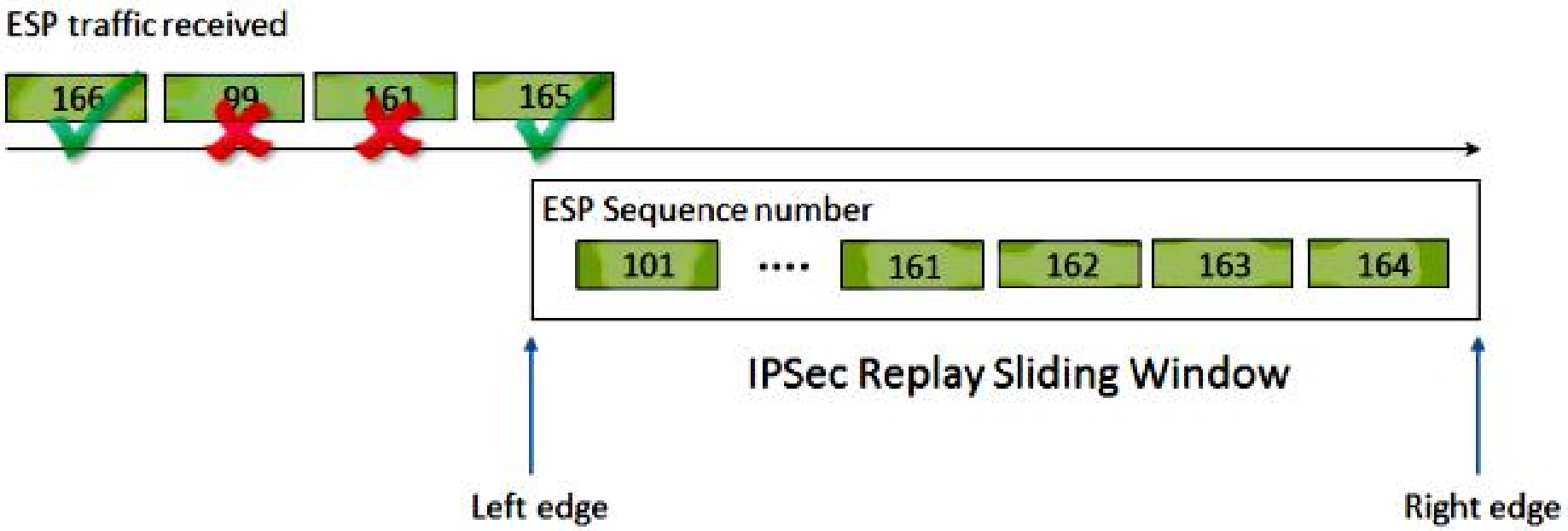
抗重放攻击：关键数据结构

```
 struct ipsec_sa
{
  ……
   u8   ipsa_replaywin; //滑动窗口大小
   //最后一个包的序号
   u32 ipsa_replaywin_lastseq;
   //已接收包位图，记录所有已接收过的序号
   u32 ipsa_replaywin_bitmap;
   //最大包序号之间差值
   u32 ipsa_replaywin_maxdiff;
   u32 ipsa_replaywin_errs ; //错误包序号
   ……
}
```

抗重放攻击：处理流程

1. 判断接收到的包序号是否**小于**最后一个包的序号，如果是，再检查该包序号与最大包序号之间差值ipsa_replaywin_maxdiff：是否大于窗口大小，如果大于则说明数据包已经过期，否则转2
2. 根据ipsa_replaywin_bitmap判断该序列号对应的包是否已经接收过，如果是则确认为重放，反之不是重放，并更新ipsa_replaywin_bitmap
3. 如果接收到的序列号比最后一个包的序号大，并且差值小于窗口大小，则窗口向右滑动，否则窗口不滑动。最后更新 ipsa_replaywin_lastseq的值为本包序列号。

Antireplay attack: example



安全参数协商

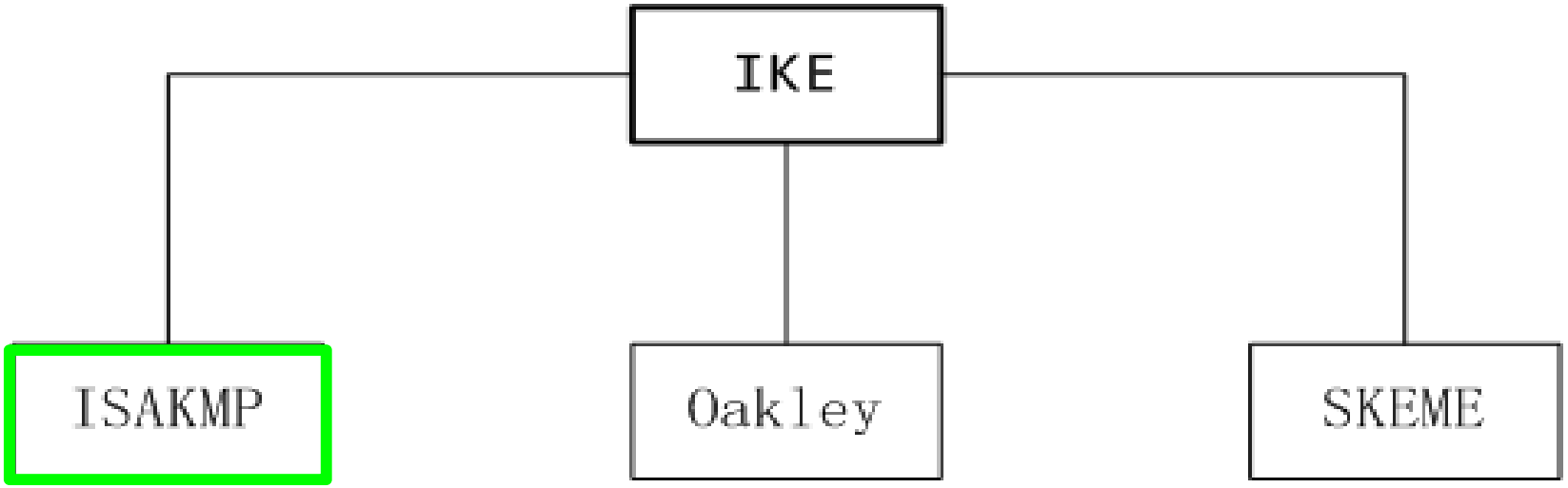
IKE

- IKE, Internet Key Exchange
- IKE的功能
  - 协商SA
  - 密钥生成
  - 身份认证

类似:
TLS Handshake protocol

- 应用层协议
  - UDP
  - Port：500

IKE



- ISAKMP：交换时序和格式
  - Internet Security Association and Key Management Protocol
- Oakley：优化DH算法
- SKEME：快速密钥更新的通用密钥交换技术

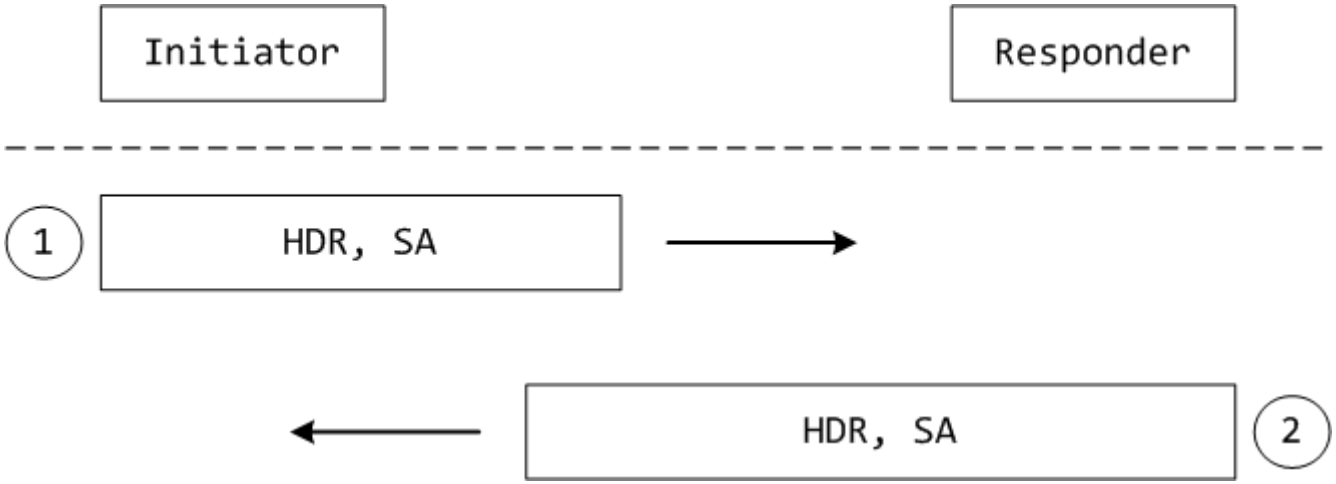
ISAKMP协商过程

- 第一阶段，协商获得ISAKMP SA，用以保护第二阶段的协商过程
- 第二阶段，协商获取安全协议SA用于保护通信数据

理解ISAKMP的两个重要概念

- 交换
  - 规定了协议消息交换的时序
  - 5种协商时序
    - 基本交换、身份保护交换、只有认证的交换、野蛮交换、通知交换
- 载荷
  - 规定了协议的语法和语义

ISAKMP：身份保护交换（1）

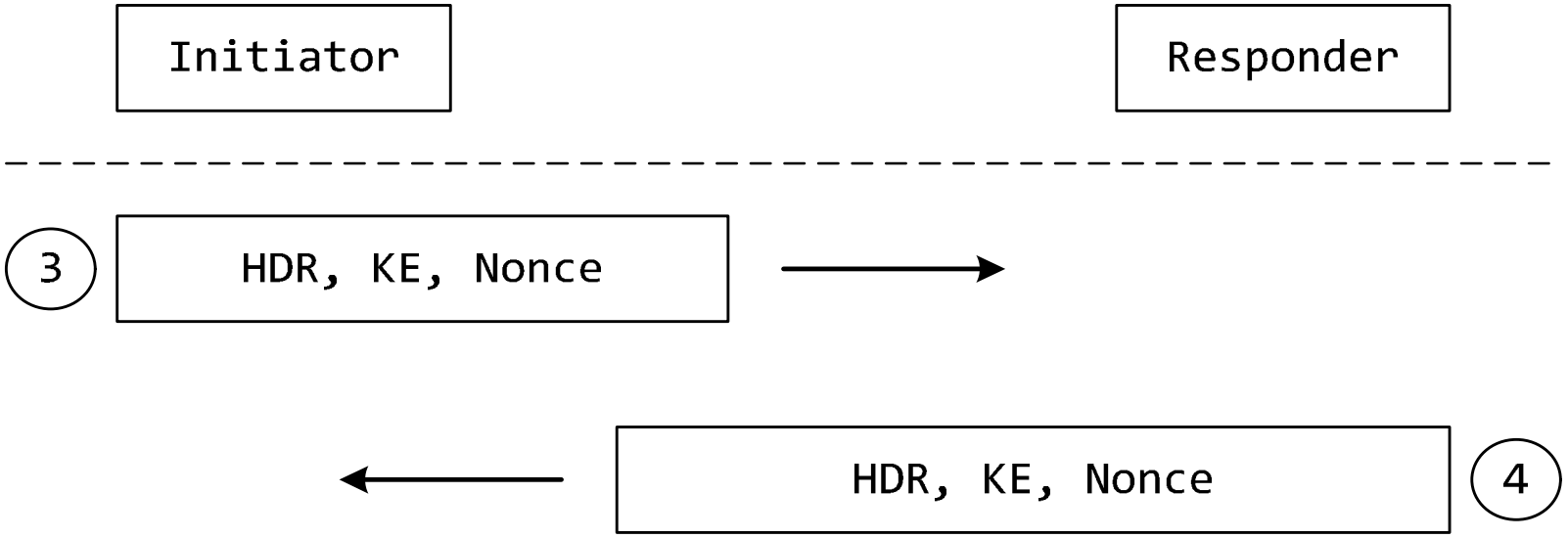


① 发起方发送第一个报文，包括首部和SA，SA是发起方给出的安全关联建议。

② 回应方回复一个报文，包括首部和回应方选择的SA。

至此，完成SA协商

ISAKMP：身份保护交换（2）

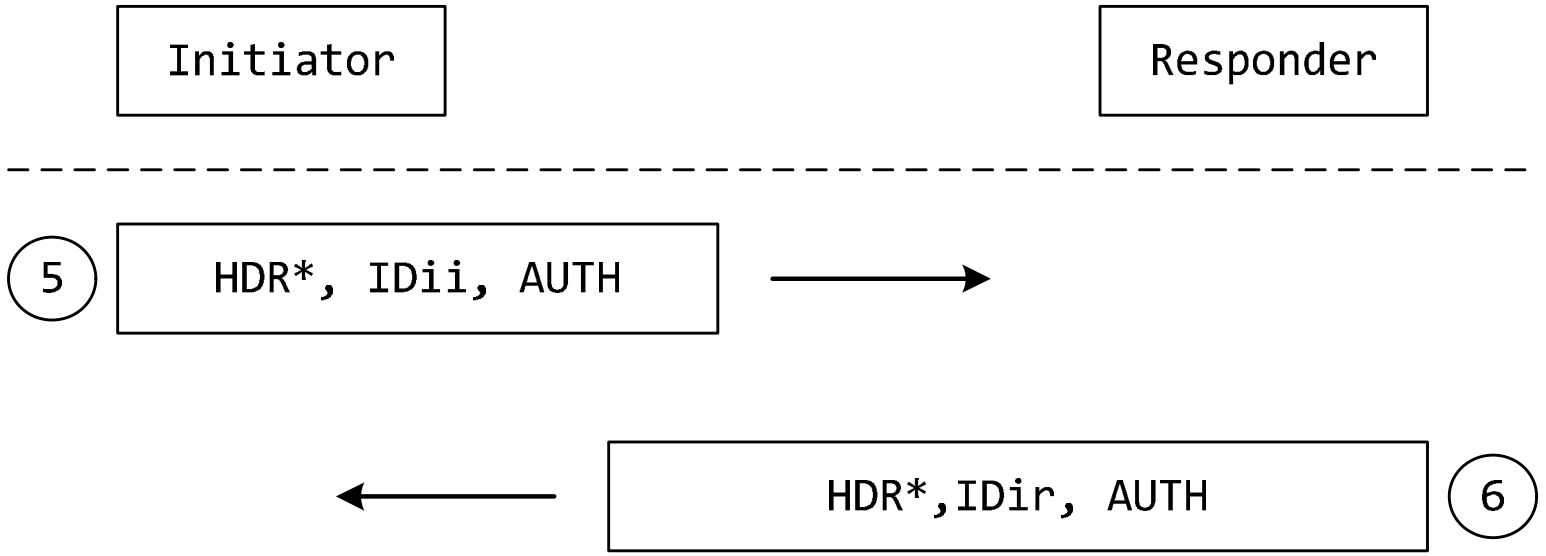


③ 发起方发送报文，包括首部、密钥交换信息和随机数

④ 回应方响应一个报文，包括首部、密钥交换信息和随机数

至此，完成密钥协商

ISAKMP：身份保护交换（3）

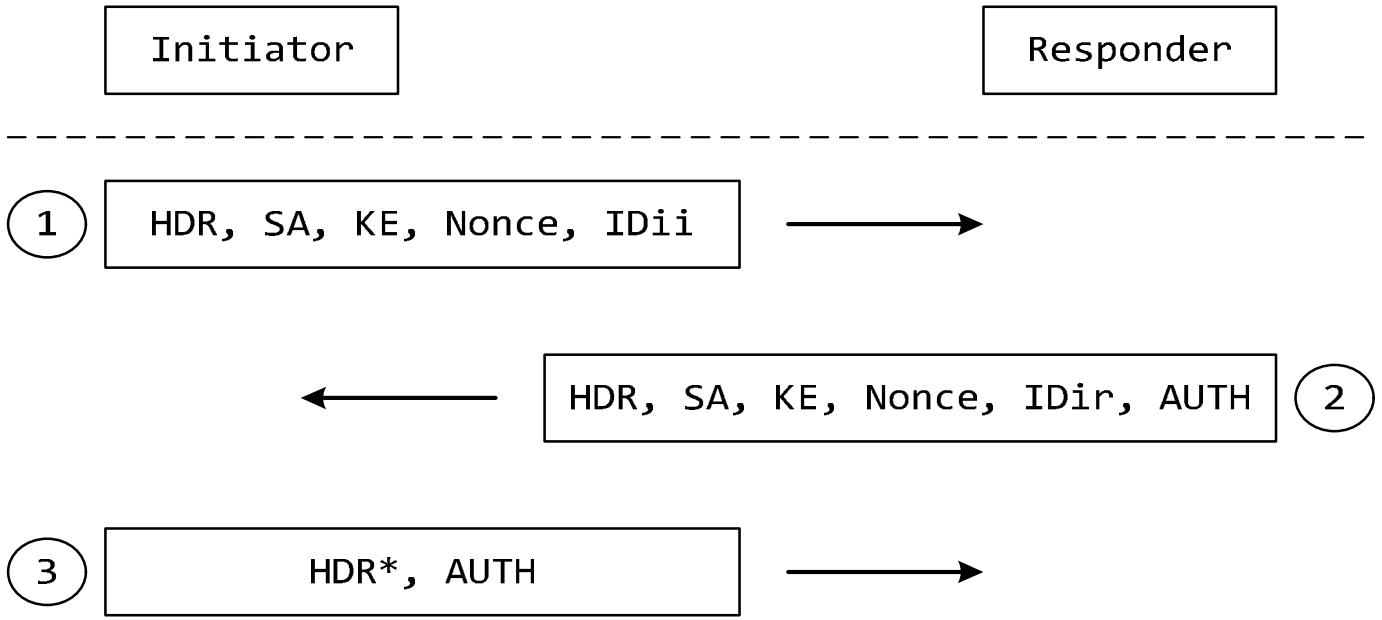


⑤ 发起方发送一个报文，包括首部、身份信息和认证信息，**数据区受加密保护**。

⑥ 回应方回复一个报文，包括首部、身份信息和认证信息，**数据区受加密保护**。

至此，完成双方身份认证

ISAKMP: Aggressive Exchange



① 发起方发送一个报文，包括首部、SA、KE、Nonce和IDii

② 回应方回复一个报文，包括首部、选定的SA、KE、Nonce、IDir、AUTH

③ 发起方发送一个报文，包括首部和认证信息，**认证信息加密**

ISAKMP:通知交换

- 用途：
  - 在ISAKMP交换过程中，如果某一方发现有差错发生，则用“通知交换”通告通信对端
  - 用于SA管理，比如通知通信对端删除某个SA
- 通知交换报文用第一阶段协商的安全参数进行保护

ISAKMP：报文及载荷

由载荷构成

取决于交换类型及报文类型

ISAKMP：报文首部

报文首部字段说明（1）

- I-Cookie：8B，用以标识ISAKMP SA
- R-Cookie：8B，用以标识ISAKMP SA
- 下一载荷：1B，说明报文的第一个载荷类型
- 主版本号和次版本号：分别为4b,说明所使用的ISAKMP版本
- 交换类型：1B，说明所使用的交换类型，如：身份保护交换（2），野蛮交换（4）等

报文首部字段说明（2）

- 标志：字段长度1B，前5个比特固定为0，后面3个比特分别为A比特、C比特、E比特
  - E比特：加密位。如果该比特为1，则说明首部之后的数据采用了ISAKMP SA所指定的加密算法进行了加密
  - C比特：同步位。在通信对端发送的C比特为1的通知交换报文时，说明通信双方的SA协商已经完成。
  - A比特：认证位。如果该比特为1，则报文的数据区仅包含认证信息，并未做加密处理。

报文首部字段说明（3）

- Message ID：发起方生成的一个4B随机数。在第一阶段协商中该字段必须设置为0；在第二阶段协商中，与SPI一起标识SA。
- 长度：指明包括首部在内的整个报文的字节数。

ISAKMP报文：下一载荷

- 指明报文中第一个载荷的类型
- 载荷类型：
  - Security Association（SA）：值为1
  - Proposal（P）：值为2
  - Transform（T）：值为3
  - Key Exchange（KE）：值为4
  - Hash（HASH）：值为8
  - Signature（SIG）：值为9
  - Nonce（NONCE）：值为10
  - ……

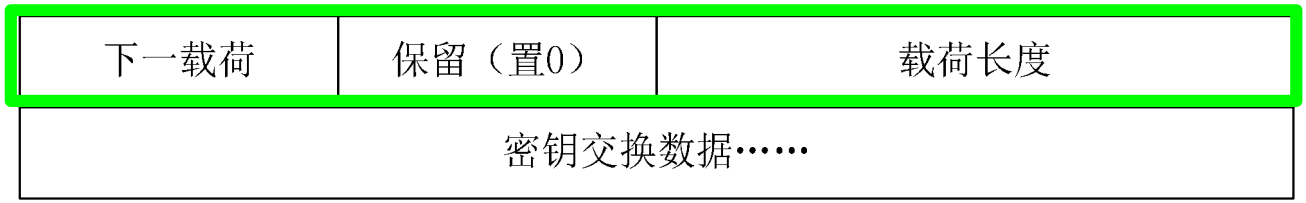
ISAKMP：载荷

- **通用载荷首部 + 载荷类型相应的数据**
- 通用载荷首部：

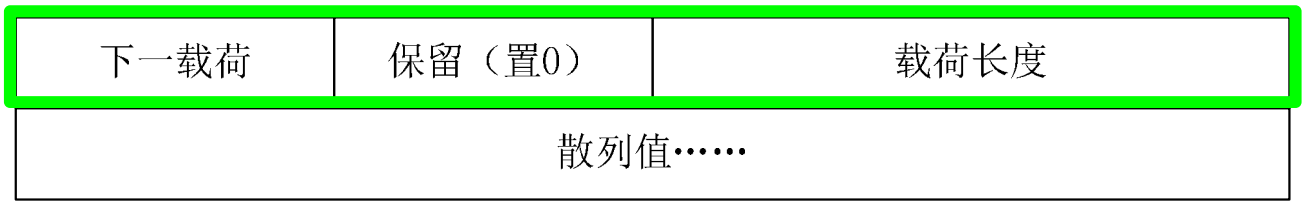
下一载荷	保留（置0）	载荷长度

- 下一载荷：1B，说明随后一个载荷的类型；
- 保留：1B，设置为0；
- 载荷长度：2B，包括载荷首部在内的整个载荷所占的字节数。

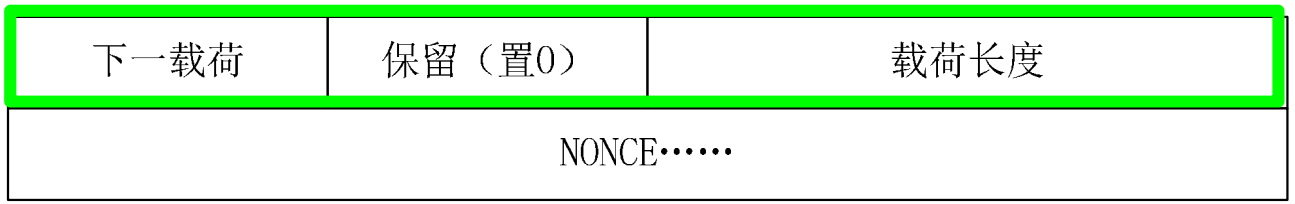
KE、HASH、NONCE载荷



密钥交换载荷格式（KE）



散列载荷格式（HASH）

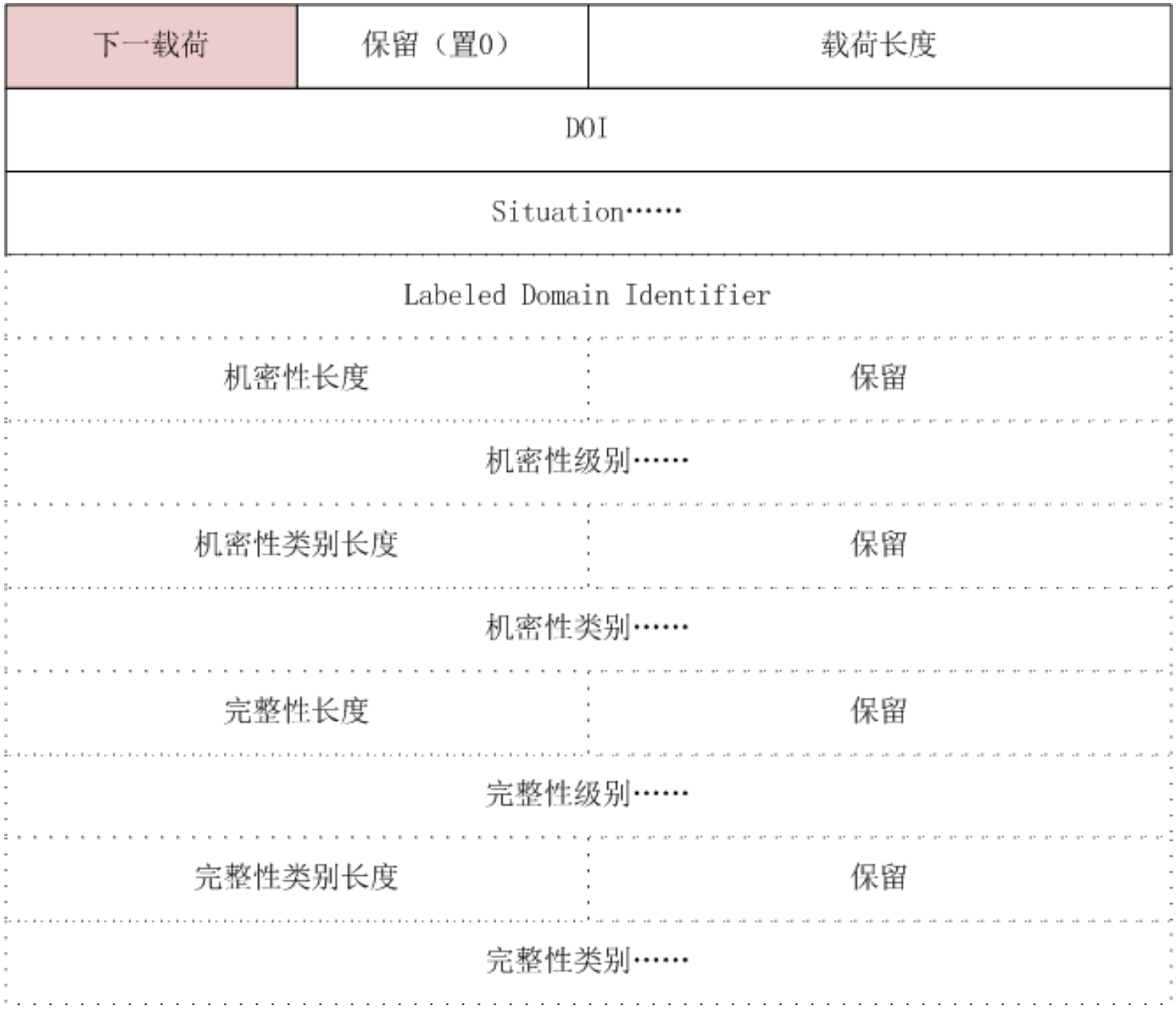


NONCE载荷格式

SA载荷

- 在协商过程中，发起方通过SA载荷向回应方提议安全方案，从而满足安全需求。
- 发起方可以提供多套方案，每套方案用一个P载荷体现，T载荷则描述安全方案的细节
- SA、P和T载荷应结合使用来协商SA
- T载荷属于P载荷，而P载荷属于SA载荷。

SA 载荷格式


下一载荷
保留（置0）
载荷长度
DOI
Situation……
Labeled Domain Identifier
机密性长度
保留
机密性级别……
机密性类别长度
保留
机密性类别……
完整性长度
保留
完整性级别……
完整性类别长度
保留
完整性类别……

SA载荷字段说明

- DOI:解释域
  - 不同应用环境对安全需求的描述不同，因此需要给定一个具体的环境来解释，即解释域
  - 4B，第一阶段协商中，取值0；第二阶段协商中，取值1，表示IPSec解释域
- Situation：4B，用于描述安全需求，具体取值与DOI有关。
- 其余字段可选

P载荷格式

下一载荷	保留（置0）	载荷长度	
Proposal#	Protocol ID	SPI长度	Number of Transforms
SPI			

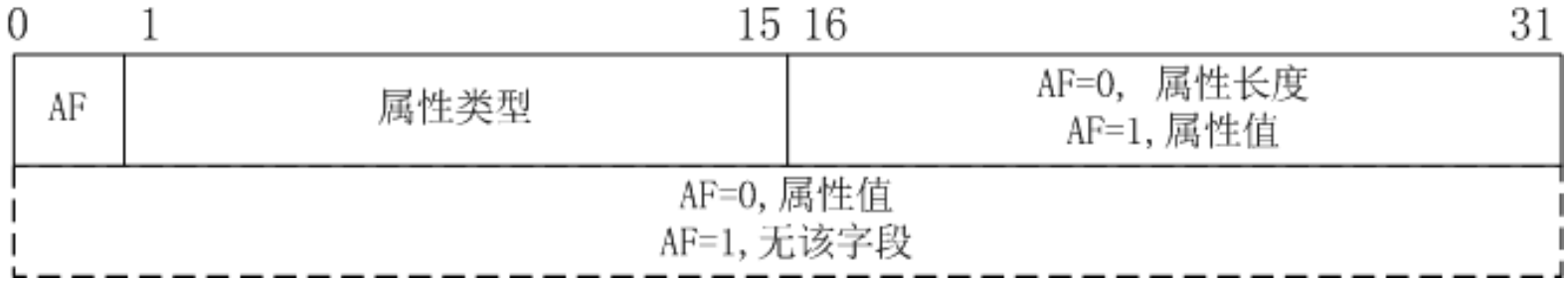
- Proposal#:1B,说明当前建议的编号，当多个P载荷属于同一套建议时，它们的编号相同
- Protocol ID：1B,说明当前P载荷对应的安全协议，ISAKMP的协议编号为1，AH为2，ESP为3.
- Transform数量：说明当前P载荷包含的T载荷数量。
- SPI：用于标识安全协议SA，长度可变，与“SPI长度”字段结合使用

T载荷格式

下一载荷	保留（置0）	载荷长度
Transform#	Transform ID	保留（置0）
SA属性		

- Transform#:1B,说明当前T载荷的编号
- Transform ID：1B，说明当前T载荷所服务的安全协议以及相应算法，ISAKMP对安全协议和相应算法有规定的Transform ID值。比如：AH_SHA的Transform ID为3，ESP_DES的Transform ID为2

T载荷：SA属性格式



- AF比特：表明SA属性的描述方式
  - AF=1，表明为短格式方式，“属性类型” 字段说明属性的内容，占用15bit，随后2个字节描述属性值
  - AF=0，表明为长格式方式，“属性类型” 字段说明属性的内容，占用15bit，随后2个字节说明属性长度，即随后的属性值字段的字节数。

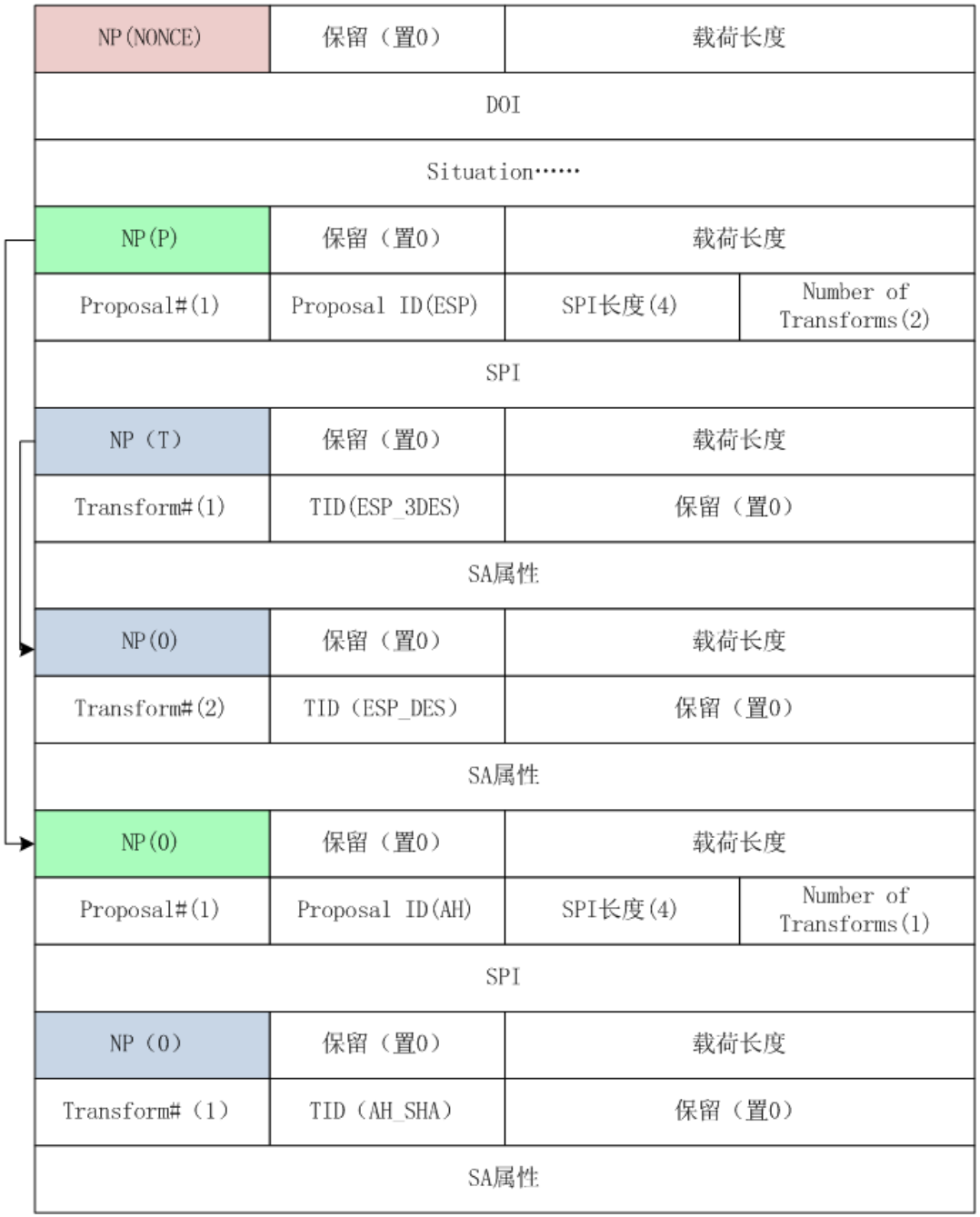
T载荷：SA属性解析举例

- 属性：0x80010001
  - 二进制表示：1000 0000 0000 0001 0000 0000 0000 0001
  - AF比特为1，表明为短格式描述方式；属性类型值为1，表明为“SA生存期类型”；属性值为1，表明以“秒”为生存期计算单位
- 属性：0x00020004 00015180
  - 二进制表示：0000 0000 0000 0010 0000 0000 0000 0100 0000 0000 0000 0001 0101 0001 1000 0000
  - AF比特为0，表示长格式描述方式；属性类型值为2，表明属性类别为“SA生存期”；属性值长度为4，说明后面的属性值要占4个字节；属性值为0X00015180，即这个SA的生存期为24小时。

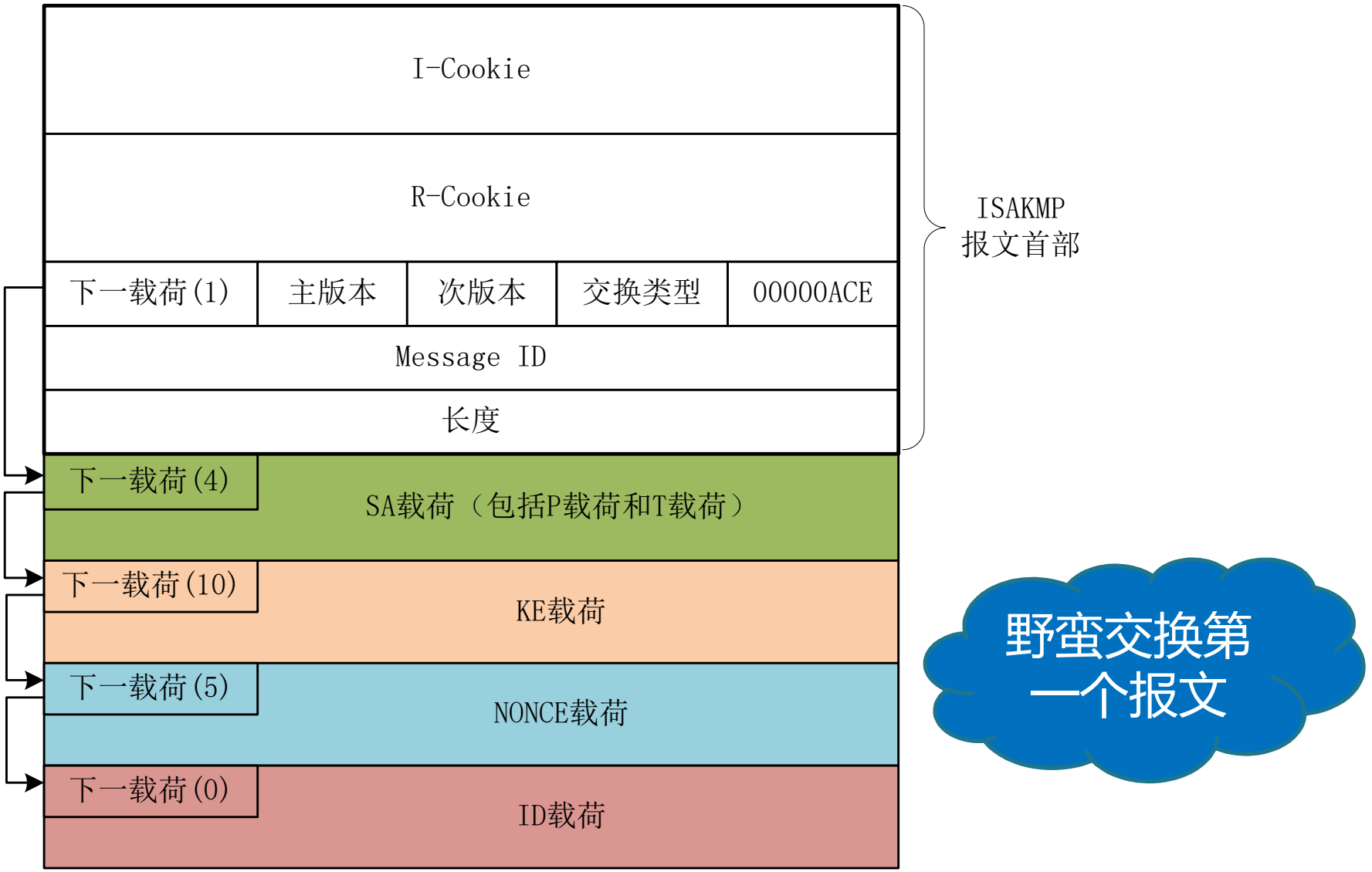
T载荷：SA属性解析举例

- 属性：0x80010002
  - 二进制表示：1000 0000 0000 0001 0000 0000 0000 0010
  - AF比特为1，表明为短格式描述方式；属性类型值为1，表明为“SA生存期类型”，属性值为2，表明生存期以处理过的数据量（kB）来计算
- 属性：0x00020004 000186A0
  - 二进制表示：0000 0000 0000 0010 0000 0000 0000 0100 0000 0000 0000 0001 0101 0001 1000 0000
  - AF比特为0，表示长格式描述方式；属性类型值为2，表明属性类别为“SA生存期”；属性值长度为4，说明后面的属性值要占4个字节；属性值为0X000186A0，即这个SA的生存期为处理100MB数据。

SA 协商载荷示例


NP(NONCE)
保留（置0）
载荷长度
DOI
Situation……
NP(P)
保留（置0）
载荷长度
Proposal#(1)
Proposal ID(ESP)
SPI长度(4)
Number of Transforms(2)
SPI
NP（T）
保留（置0）
载荷长度
Transform#(1)
TID(ESP_3DES)
保留（置0）
SA属性
NP(0)
保留（置0）
载荷长度
Transform#(2)
TID（ESP_DES）
保留（置0）
SA属性
NP(0)
保留（置0）
载荷长度
Proposal#(1)
Proposal ID(AH)
SPI长度(4)
Number of Transforms(1)
SPI
NP（0）
保留（置0）
载荷长度
Transform#（1）
TID（AH_SHA）
保留（置0）
SA属性

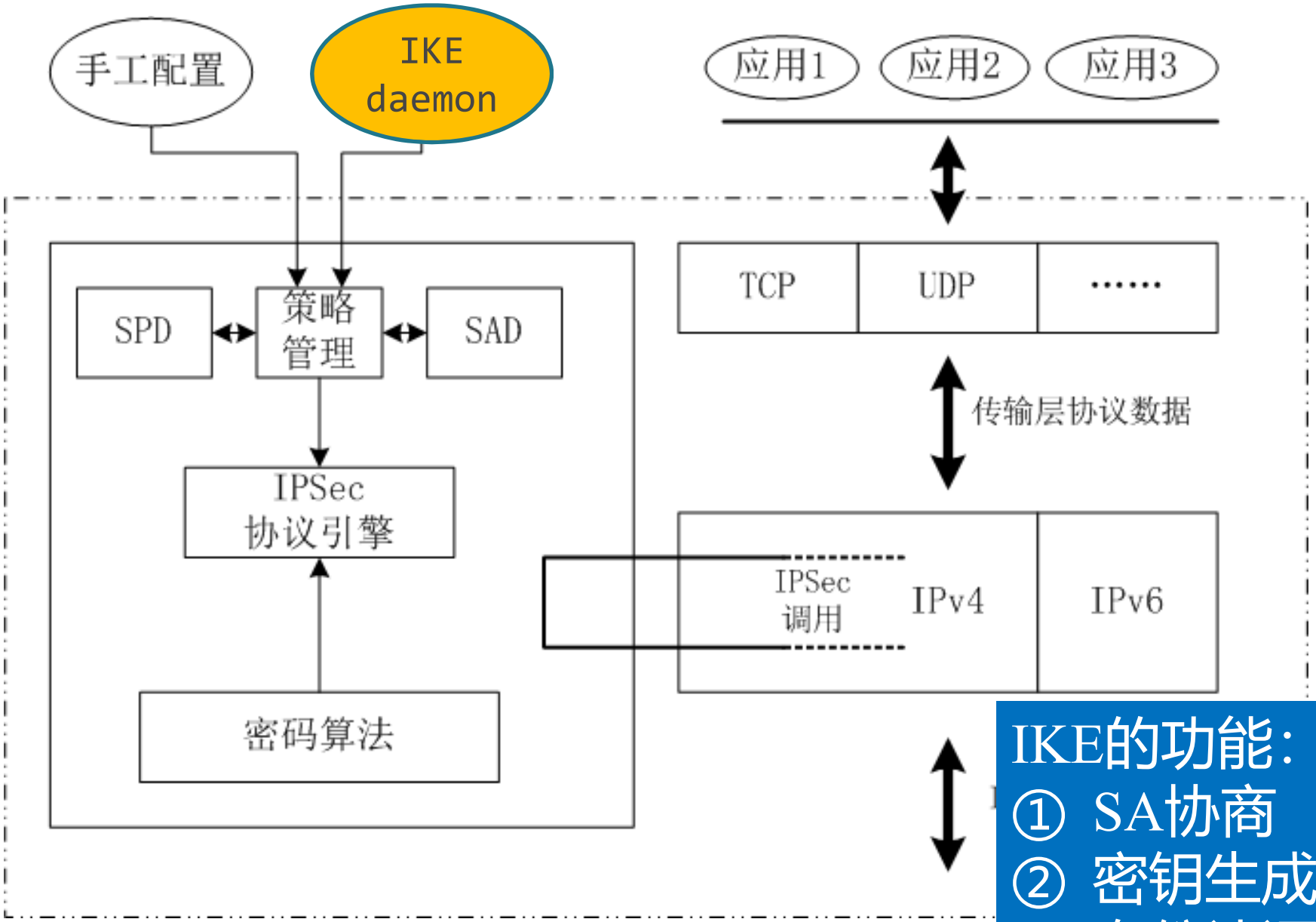
一个完整ISAKMP报文示例



ISAKMP总结

- 定义了协商的流程
  - 5种交换
- 定义了报文格式
  - 报文首部格式
  - 各种载荷格式
- 未定义具体的认证方式
- 未定义具体的密钥生成方法

Internet Key Exchange (IKE)



IKE的功能：

① SA协商

② 密钥生成

③ 身份认证

IKEv1

- 第一阶段，协商获得IKE SA
  - 主模式（main mode）
    - 对应ISAKMP：Identity Protect Exchange
  - 野蛮模式（aggressive mode）
    - 对应ISAKMP：Aggressive Exchange
- 第二阶段，协商获得IPSec SA
  - 快速模式

SA协商的SA属性（1）

- 加密算法
  - DES-CBC, IDEA-CBC, 3DES-CBC,……
- 散列算法
  - MD5, SHA, Tiger， ……
- 认证方法
  - DSS签名，RSA加密， 预共享密钥，……

SA协商的SA属性（2）

- DH群类型
  - MODP, EC2N, ECP
- 伪随机函数
- 密钥长度
- 生命期类型及生命期
- ……

SA协商属性示例

- IKE Attribute (t=1,l=2): Encryption-Algorithm: AES-CBC
- IKE Attribute (t=14,l=2): Key-Length: 128
- IKE Attribute (t=2,l=2): Hash-Algorithm: SHA
- IKE Attribute (t=4,l=2): Group-Description: 2048 bit MODP group
- IKE Attribute (t=3,l=2): Authentication-Method: Pre-shared key
- IKE Attribute (t=11,l=2): Life-Type: Seconds
- IKE Attribute (t=12,l=2): Life-Duration: 3600

SA属性：伪随机函数（PRF）

- 伪随机函数：以秘密信息和其他信息作为输入，产生随机的比特流
- IKE使用PRF生成4种秘密信息：
  - SKEYID：用于导出其它秘密信息
  - SKEYID_d：为IPSec衍生出加密的密钥素材
  - SKEYID_a：用于数据完整性检验及数据源发认证
  - SKEYID_e：用于数据加密

SA属性：伪随机函数（PRF）

- 如果不协商PRF，则默认使用HMAC，其中的hash函数为通信双方协商SA时所选定的算法
- SKEYID的生成方式**取决于认证方法**

```
For Signature: SKEYID=
prf( Ni_b|Nr_b, g^xy )
```

```
For Public key encryption: SKEYID=
prf( hash(Ni_b|Nr_b), CKY_I|CKY_R)
```

```
For pre-shared Key: SKEYID=
prf( pre-shared key, Ni_b|Nr_b)
```

密钥素材的导出

- 在已知SKEYID的情况下，密钥素材按如下方式导出（与认证方法无关）：

```
SKEYID_d =
prf( SKEYID, g^xy|CKY_I|CKY_R|0 )
```

```
SKEYID_a =
prf(SKEYID,SKEYID_d|g^xy|CKY_I|CKY_R|1)
```

```
SKEYID_e =
prf(SKEYID,SKEYID_a|g^xy|CKY_I|CKY_R|2)
```

IKE: 身份认证方法

- SKEYID的导出与**身份认证方法有关**
- 第一阶段的IKE SA协商报文与**身份认证有关**
- IKE支持四种身份认证方法
  - 使用数字签名认证方法
  - 使用公钥加密的认证方法
  - 使用改进的公钥加密的认证方法
  - 使用共享密钥的认证方法

IKE: 身份认证方法（1）

- 基于数字签名的身份认证
  - 通信双方互相交换证书和签名信息，如果签名验证通过，则说明对方拥有与证书所包含公钥对应的私钥，从而确认对方身份。

```
For Signature: SKEYID=
prf( Ni_b|Nr_b, g^xy )
```

IKE: 身份认证方法（2）

- 基于公钥加密的身份认证
  - 通信双方用对方的公钥对身份、随机数Nonce等信息进行加密处理，将结果发送给对方；之后，通信双方要将身份、随机数等信息作为输入生成认证信息。如果认证信息正确，表明对方拥有公钥所对应的私钥，从而验证对端身份。

```
For Public key encryption: SKEYID=
prf( hash(Ni_b|Nr_b), CKY_I|CKY_R)
```

IKE: 身份认证方法（3）

- 改进的基于公钥加密的身份认证
  - 对基于公钥加密的身份认证方法的改进，对部分信息采用公钥进行加密，而对另一部分信息采用对称密码进行加密。在要加密的信息较多的情况下，此方法的处理效率较高。

```
For Public key encryption: SKEYID=
prf( hash(Ni_b|Nr_b), CKY_I|CKY_R)
```

IKE: 身份认证方法（4）

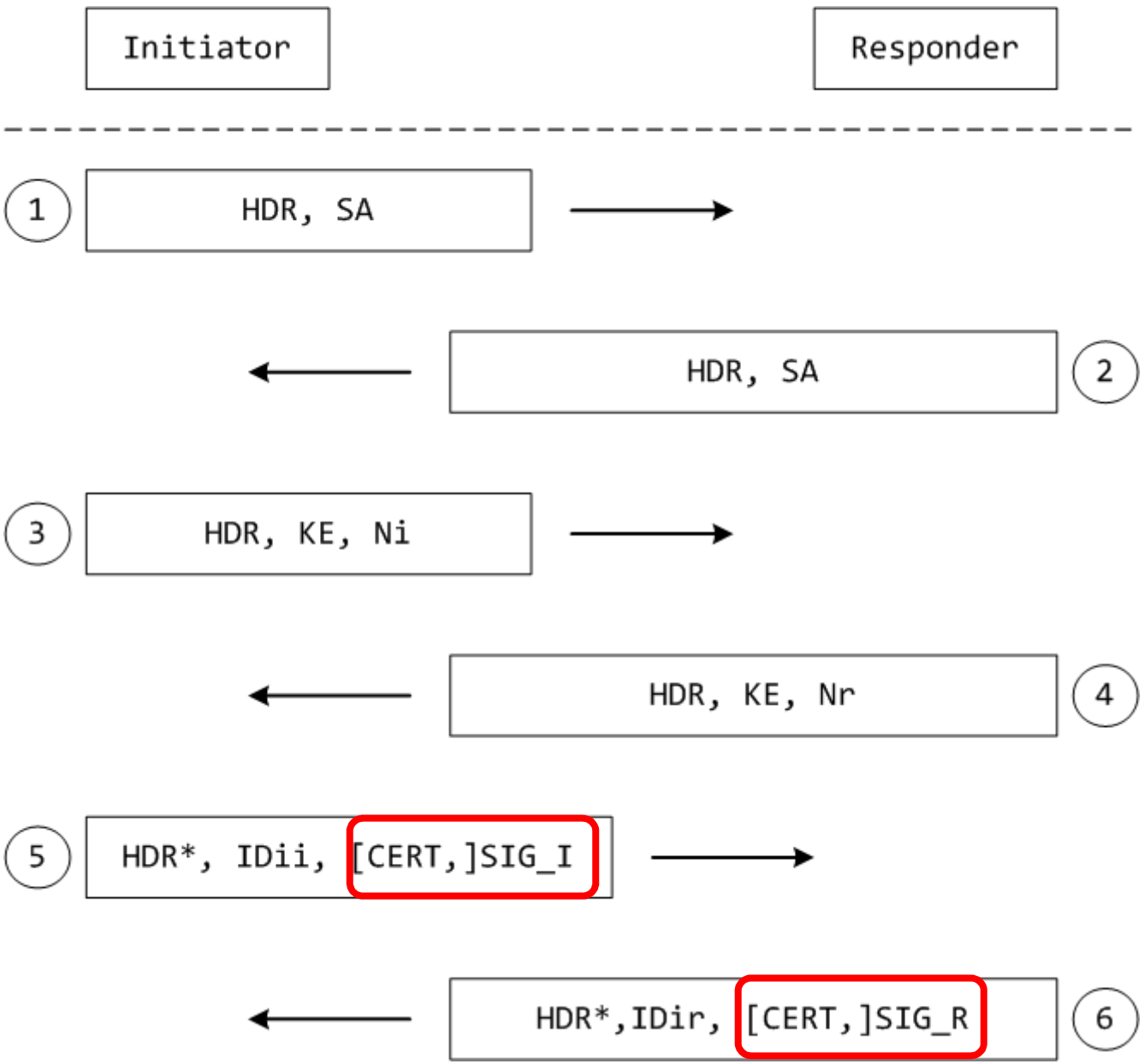
- 基于预共享密钥的身份认证
  - 此方法要求通信双方预先共享一个密钥，在生成认证信息时，预共享密钥作为输入之一；如果认证信息正确，则说明对方拥有正确的预共享密钥，从而验证对端身份。

```
For pre-shared Key: SKEYID=
prf( pre-shared key, Ni_b|Nr_b)
```

IKEv1第一阶段

- 主模式
- 野蛮模式

主模式：使用数字签名认证方法



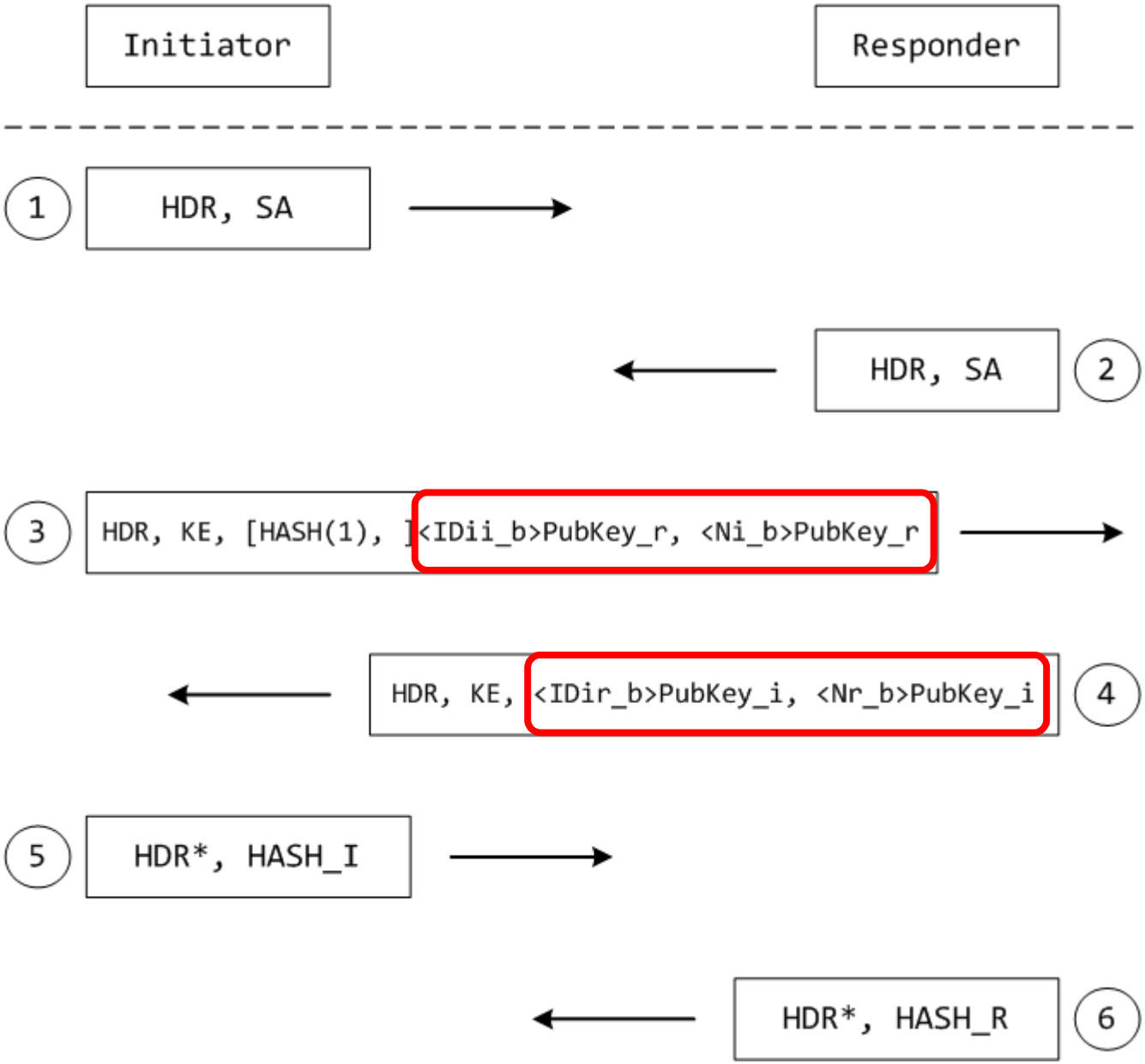
主模式：使用数字签名认证方法

- 1、2报文完成SA协商；
- 3、4报文传输密钥交换信息和随机数，完成密钥交换；
- 5、6报文进行身份和认证信息交换。

主模式：使用数字签名认证方法

- 由于是使用数字签名进行身份认证，因此，报文中包含数字签名信息，即SIG_I,SIG_R，为了验证数字签名，需要相对应的公钥，因此，报文中可能传递证书信息；证书信息也可以通过其它方式传递，因此，在这里的证书是可选的。
- SIG_I和SIG_R是通过将协商出的签名算法应用于HASH_I和HASH_R获得。

主模式：使用公钥加密认证方法



主模式：使用公钥加密认证方法

- 报文1和2协商SA
- 报文3和4进行密钥交换和用对端公钥加密的身份信息及随机数
- 可选的HASH(1)用于说明所使用的回应方公钥（回应方可能有多个公钥），其输入为所选公钥对应的证书。

主模式：使用公钥加密认证方法

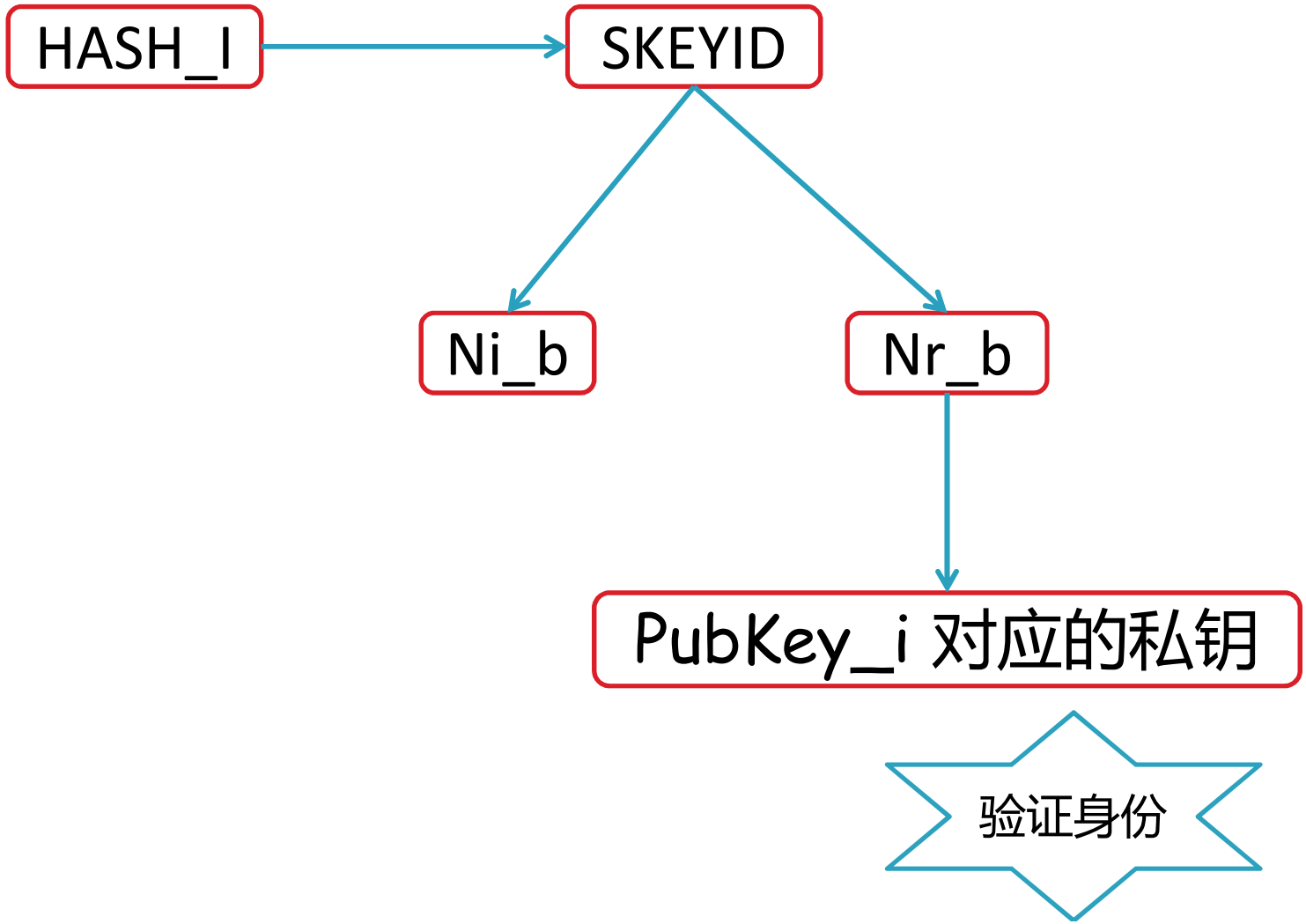
- 5、6报文的HASH_I和HASH_R用于验证对端身份。

```
HASH_I =
prf(SKEYID,g^xi|g^xr|CKY-I|CKY-R|SAi_b|IDii_b)
```

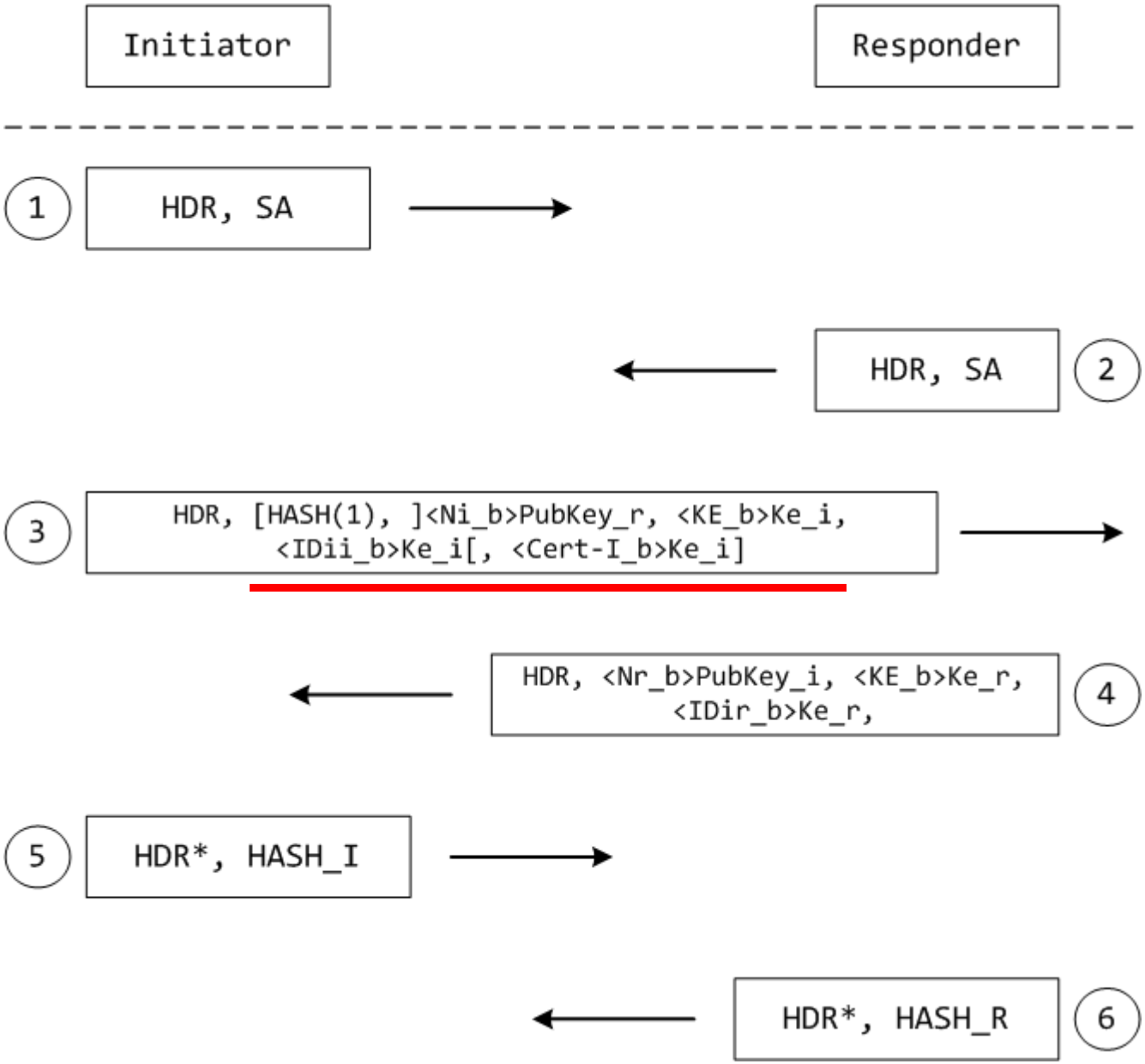
```
HASH_R =
prf(SKEYID,g^xr|g^xi|CKY-R|CKY-I|SAi_b|IDir_b)
```

```
For Public key encryption: SKEYID=
prf( hash(Ni_b|Nr_b), CKY-I|CKY-R)
```

主模式：使用公钥加密认证方法



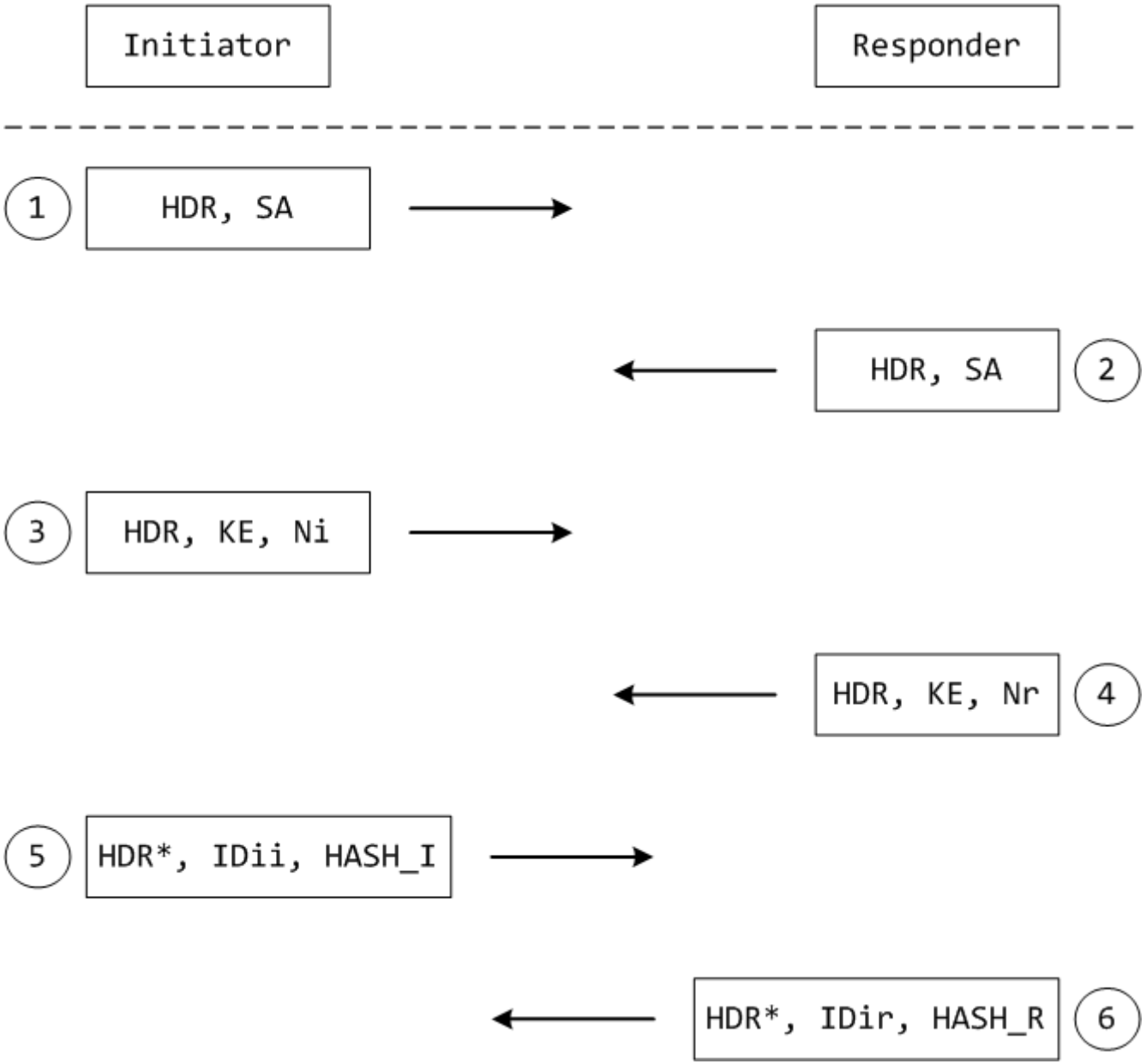
主模式：使用改进的公钥加密认证方法



主模式：使用改进的公钥加密认证方法

- 1、2报文，协商确定SA；
- 3、4报文，协商密钥交换信息和随机数、身份、证书信息（可选）。其中，随机数信息用回应方的公钥加密，而KE,ID和证书信息，用从Nonce信息导出的密钥加密，加密算法为SA中说明的加密算法。通过这种方式，可以充分利用公钥密码和对称密码的优势，提高加解密的效率。
- 5、6报文，验证对端身份

主模式：使用预共享密钥认证方法



主模式：使用预共享密钥认证方法

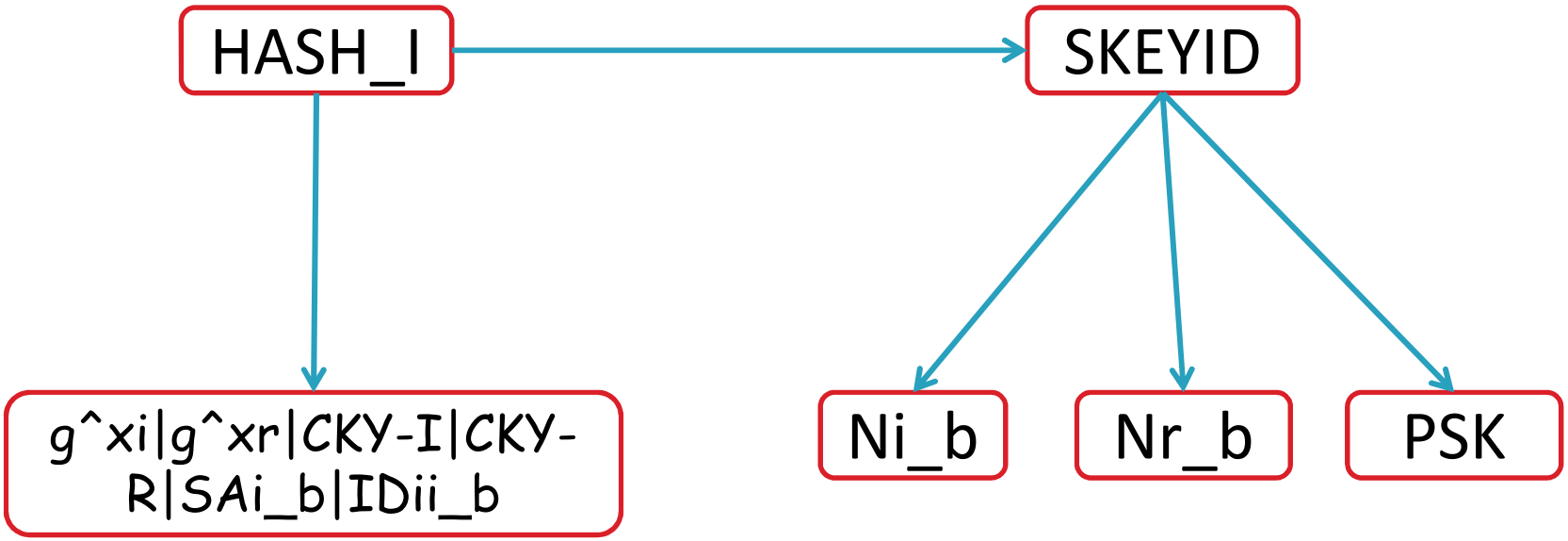
- 5、6报文的HASH_I和HASH_R用于验证对端身份和部分字段的完整性验证

```
HASH_I =
prf(SKEYID,g^xi|g^xr|CKY-I|CKY-R|SAi_b|IDii_b)
```

```
HASH_R =
prf(SKEYID,g^xr|g^xi|CKY-R|CKY-I|SAi_b|IDir_b)
```

```
For pre-shared Key: SKEYID=
prf( pre-shared key, Ni_b|Nr_b)
```

主模式：使用预共享密钥的认证方法



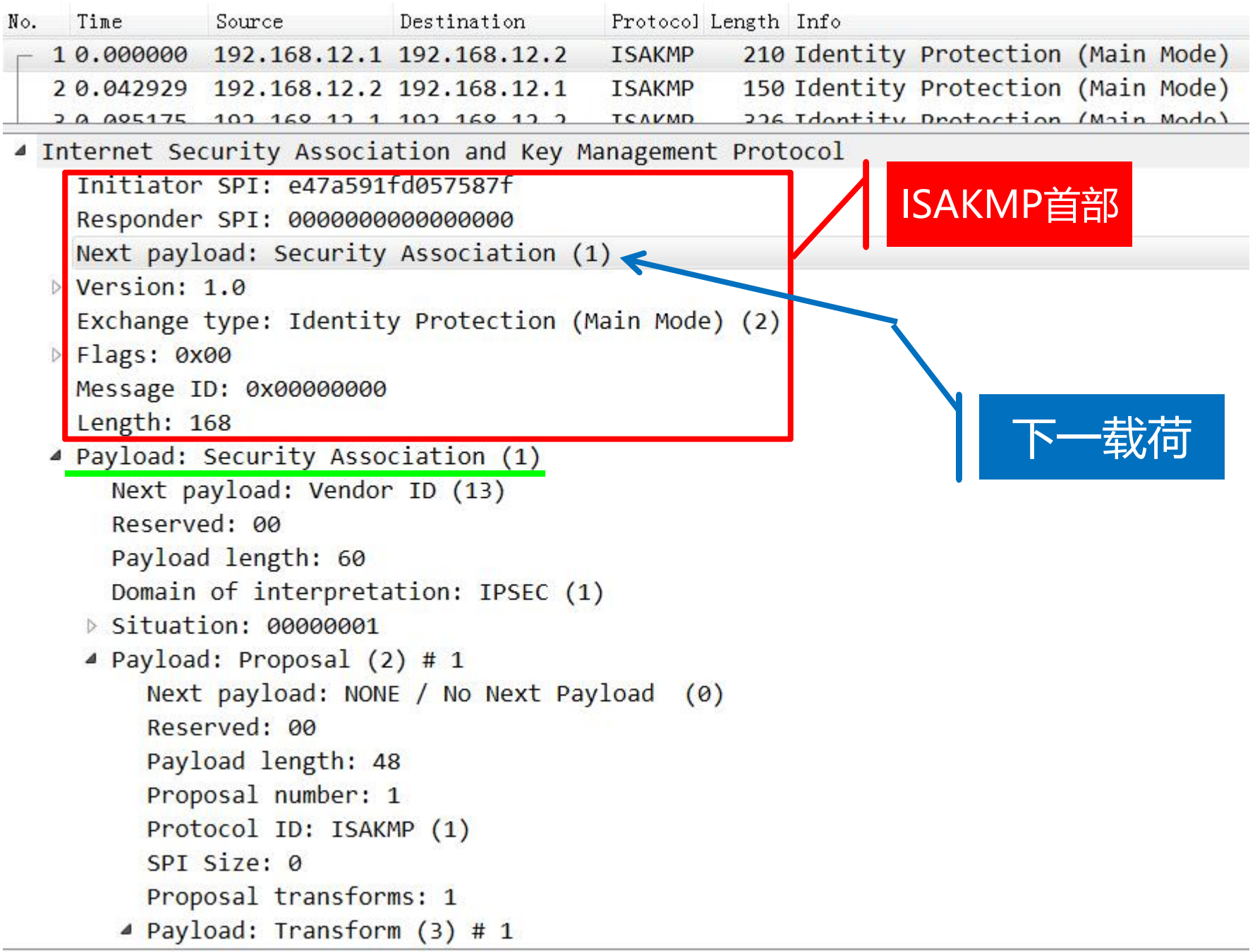
只有当对端拥有相同的预共享密钥的情况下，且Ni和Nr在传输过程中均没有被篡改的情况下，才能生成相同的SKEYID；同时，只有部分字段（prf计算里面输入）保持完整性的条件下，才能产生正确的HASH_I和HASH_R。从而验证了对端身份，也验证了部分字段的完整性

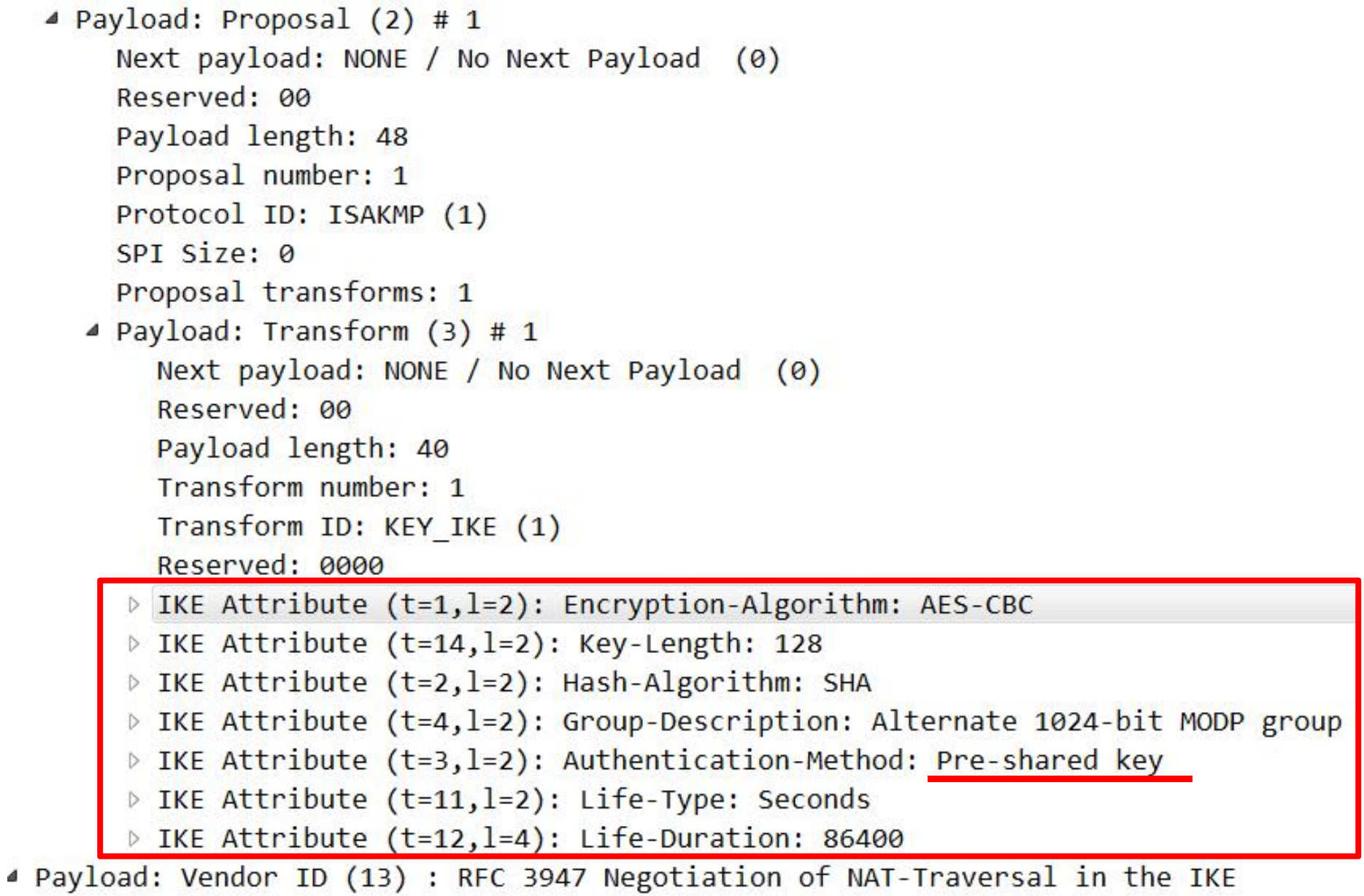
IKEv1第一阶段实例分析：第1条消息

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.12.1	192.168.12.2	ISAKMP	210	Identity Protection (Main Mode)
2	0.042929	192.168.12.2	192.168.12.1	ISAKMP	150	Identity Protection (Main Mode)
3	0.085175	192.168.12.1	192.168.12.2	ISAKMP	326	Identity Protection (Main Mode)
4	0.138292	192.168.12.2	192.168.12.1	ISAKMP	346	Identity Protection (Main Mode)
5	0.191233	192.168.12.1	192.168.12.2	ISAKMP	150	Identity Protection (Main Mode)
6	0.196275	192.168.12.2	192.168.12.1	ISAKMP	118	Identity Protection (Main Mode)
7	0.202103	192.168.12.1	192.168.12.2	ISAKMP	262	Quick Mode
8	0.208529	192.168.12.2	192.168.12.1	ISAKMP	262	Quick Mode
9	0.213251	192.168.12.1	192.168.12.2	ISAKMP	102	Quick Mode

```
▷ Frame 1: 210 bytes on wire (1680 bits), 210 bytes captured (1680 bits)
▷ Ethernet II, Src: Cisco_8b:36:d0 (00:1d:a1:8b:36:d0), Dst: Cisco_ed:7a:f0 (00:17:5a:e
▷ Internet Protocol Version 4, Src: 192.168.12.1, Dst: 192.168.12.2
▷ User Datagram Protocol, Src Port: 500, Dst Port: 500
◢ Internet Security Association and Key Management Protocol
    Initiator SPI: e47a591fd057587f
    Responder SPI: 0000000000000000
```

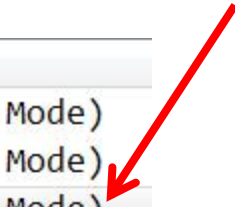
第一阶段，主模式


No. Time Source Destination Protocol Length Info
1 0.000000 192.168.12.1 192.168.12.2 ISAKMP 210 Identity Protection (Main Mode)
2 0.042929 192.168.12.2 192.168.12.1 ISAKMP 150 Identity Protection (Main Mode)
Internet Security Association and Key Management Protocol
Initiator SPI: e47a591fd057587f
Responder SPI: 0000000000000000
Next payload: Security Association (1)
Version: 1.0
Exchange type: Identity Protection (Main Mode) (2)
Flags: 0x00
Message ID: 0x00000000
Length: 168
Payload: Security Association (1)
Next payload: Vendor ID (13)
Reserved: 00
Payload length: 60
Domain of interpretation: IPSEC (1)
Situation: 00000001
Payload: Proposal (2) # 1
Next payload: NONE / No Next Payload (0)
Reserved: 00
Payload length: 48
Proposal number: 1
Protocol ID: ISAKMP (1)
SPI Size: 0
Proposal transforms: 1
Payload: Transform (3) # 1
ISAKMP首部
下一载荷


Payload: Proposal (2) # 1
Next payload: NONE / No Next Payload (0)
Reserved: 00
Payload length: 48
Proposal number: 1
Protocol ID: ISAKMP (1)
SPI Size: 0
Proposal transforms: 1
Payload: Transform (3) # 1
Next payload: NONE / No Next Payload (0)
Reserved: 00
Payload length: 40
Transform number: 1
Transform ID: KEY_IKE (1)
Reserved: 0000
IKE Attribute (t=1,l=2): Encryption-Algorithm: AES-CBC
IKE Attribute (t=14,l=2): Key-Length: 128
IKE Attribute (t=2,l=2): Hash-Algorithm: SHA
IKE Attribute (t=4,l=2): Group-Description: Alternate 1024-bit MODP group
IKE Attribute (t=3,l=2): Authentication-Method: Pre-shared key
IKE Attribute (t=11,l=2): Life-Type: Seconds
IKE Attribute (t=12,l=4): Life-Duration: 86400
Payload: Vendor ID (13) : RFC 3947 Negotiation of NAT-Traversal in the IKE

第三条消息

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.12.1	192.168.12.2	ISAKMP	210	Identity Protection (Main Mode)
2	0.042929	192.168.12.2	192.168.12.1	ISAKMP	150	Identity Protection (Main Mode)
3	0.085175	192.168.12.1	192.168.12.2	ISAKMP	326	Identity Protection (Main Mode)
4	0.138292	192.168.12.2	192.168.12.1	ISAKMP	346	Identity Protection (Main Mode)



```
▷ User Datagram Protocol, Src Port: 500, Dst Port: 500
◢ Internet Security Association and Key Management Protocol
    Initiator SPI: e47a591fd057587f
    Responder SPI: a00b8ef0902bb8ec
    Next payload: Key Exchange (4)
  ▷ Version: 1.0
    Exchange type: Identity Protection (Main Mode) (2)
  ▷ Flags: 0x00
    Message ID: 0x00000000
    Length: 284
  ◢ Payload: Key Exchange (4)
      Next payload: Nonce (10)
      Reserved: 00
      Payload length: 132
      Key Exchange Data: 3504d3d2ed14e0ca03b851a51a9da2e5a4c14c1d7ec3e1fb...
  ◢ Payload: Nonce (10)
      Next payload: Vendor ID (13)
      Reserved: 00
      Payload length: 24
      Nonce DATA: 89d7c8fbf94b515b521d5d9589c2602021e1a709
  ▷ Payload: Vendor ID (13) : RFC 3706 DPD (Dead Peer Detection)
```

第五条消息

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.12.1	192.168.12.2	ISAKMP	210	Identity Protection (Main Mode)
2	0.042929	192.168.12.2	192.168.12.1	ISAKMP	150	Identity Protection (Main Mode)
3	0.085175	192.168.12.1	192.168.12.2	ISAKMP	326	Identity Protection (Main Mode)
4	0.138292	192.168.12.2	192.168.12.1	ISAKMP	346	Identity Protection (Main Mode)
5	0.191233	192.168.12.1	192.168.12.2	ISAKMP	150	Identity Protection (Main Mode)
6	0.196275	192.168.12.2	192.168.12.1	ISAKMP	118	Identity Protection (Main Mode)
7	0.202103	192.168.12.1	192.168.12.2	ISAKMP	262	Quick Mode

```
▷ Frame 5: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits)
▷ Ethernet II, Src: Cisco_8b:36:d0 (00:1d:a1:8b:36:d0), Dst: Cisco_ed:7a:f0 (00:17:5a:ed:
▷ Internet Protocol Version 4, Src: 192.168.12.1, Dst: 192.168.12.2
▷ User Datagram Protocol, Src Port: 500, Dst Port: 500
◢ Internet Security Association and Key Management Protocol
    Initiator SPI: e47a591fd057587f
    Responder SPI: a00b8ef0902bb8ec
    Next payload: Identification (5)
  ▷ Version: 1.0
    Exchange type: Identity Protection (Main Mode) (2)
  ▷ Flags: 0x01
    Message ID: 0x00000000
    Length: 108
    Encrypted Data (80 bytes)
```

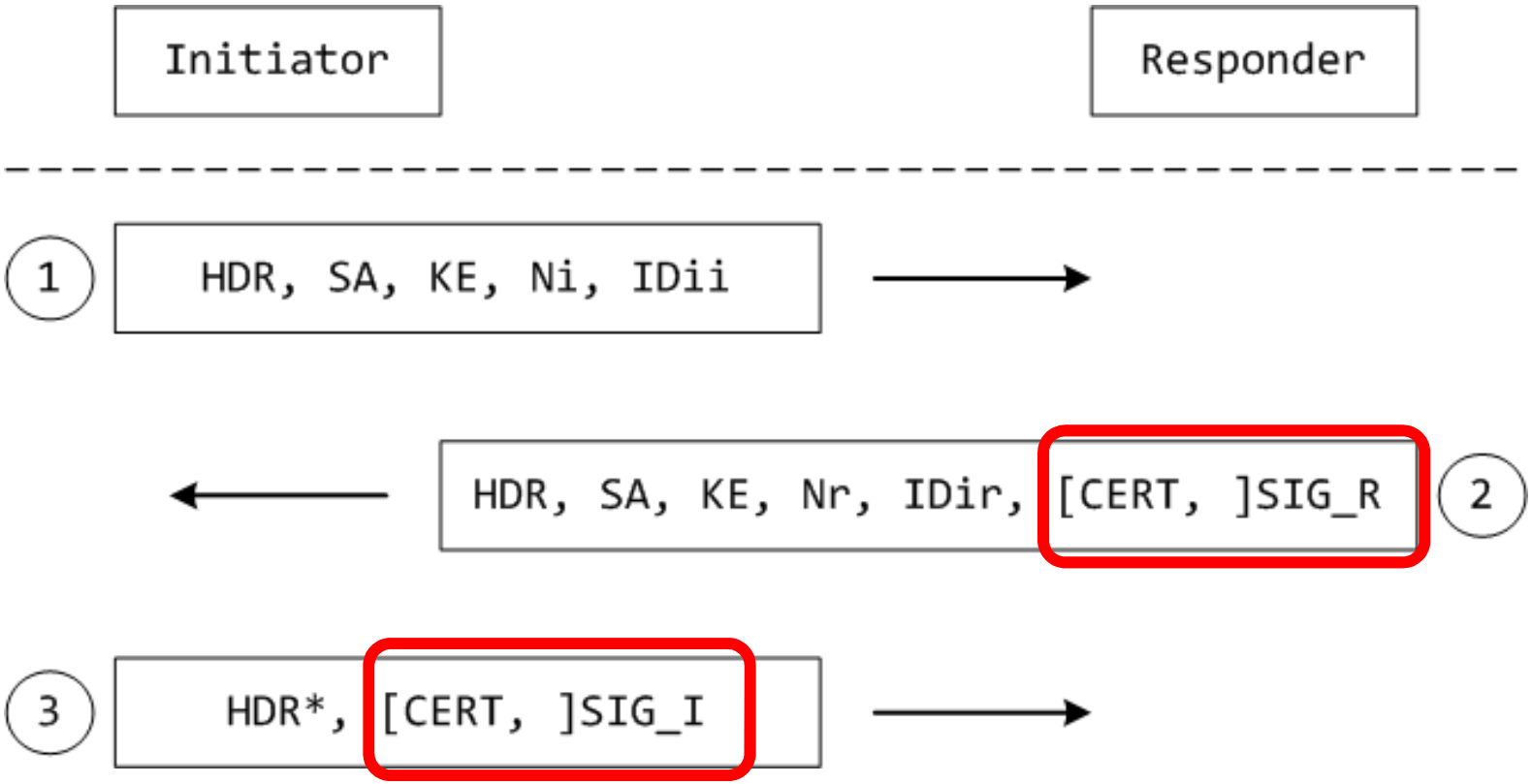
HDR后面是加密的数据

IKEv1第一阶段：野蛮模式

IKEv1第一阶段：野蛮模式

- 基于ISAKMP的野蛮交换
- 协议细节依赖于身份认证方法
- 传输内容与主模式相似，但是报文数量和报文内容不同

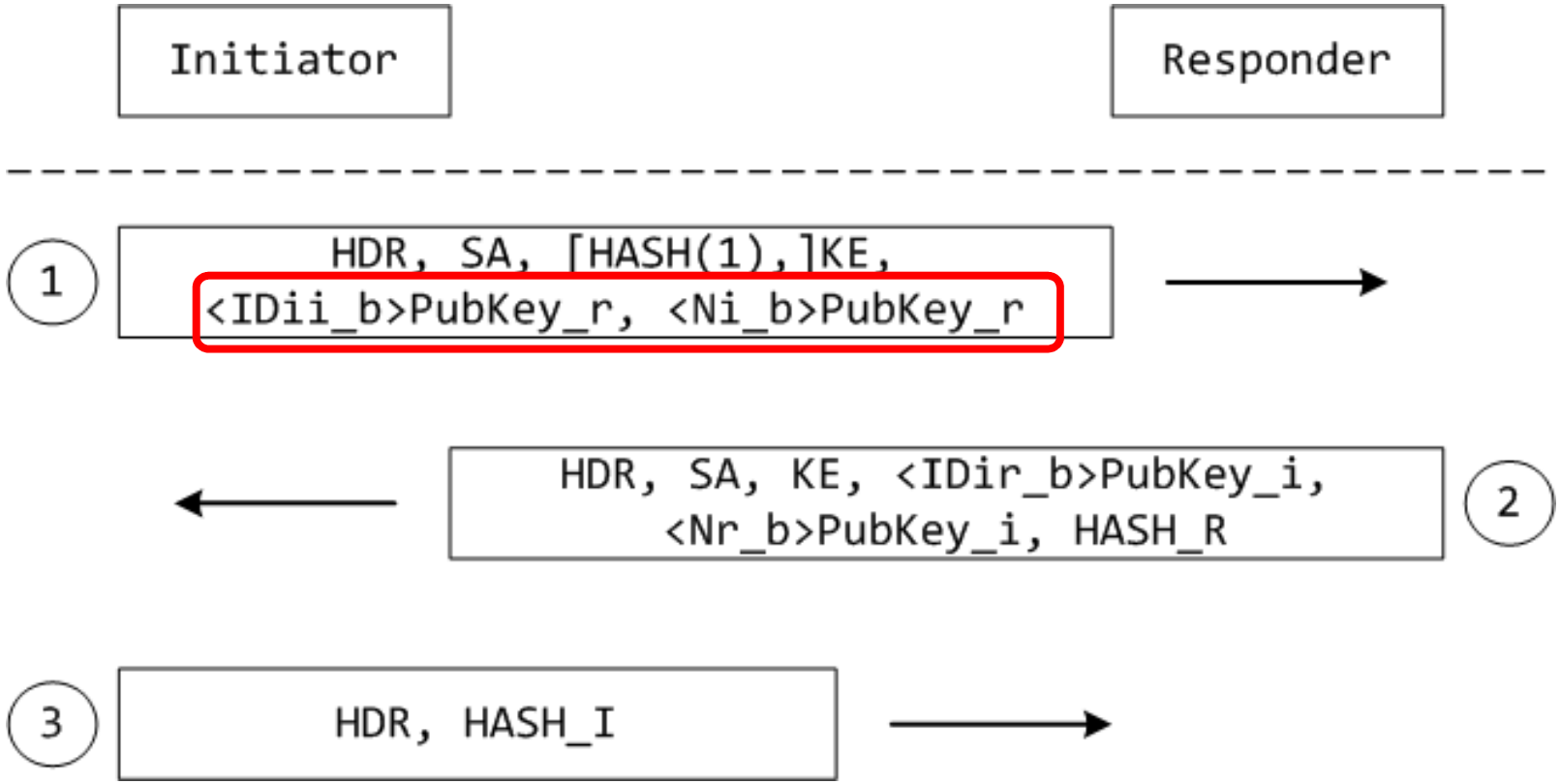
野蛮模式：使用数字签名


Initiator
Responder
1
HDR, SA, KE, Ni, IDii
HDR, SA, KE, Nr, IDir, [CERT,]SIG_R
2
3
HDR*, [CERT,]SIG_I

野蛮模式：使用数字签名

- 第1、2个报文完成SA协商，交换了KE、Nonce信息、身份信息；报文2还提供了签名信息和可选的数字证书，从而发起方可以验证回应方的身份。
- 报文3利用报文1、2协商确定的SA和密钥，对数据区进行了加密，也就是发起方的证书信息和签名信息被加密了。回应方利用相同的加密算法和密钥解密数据区，从而可以验证发起方的身份。
- 最终完成SA协商、密钥交换和身份认证。

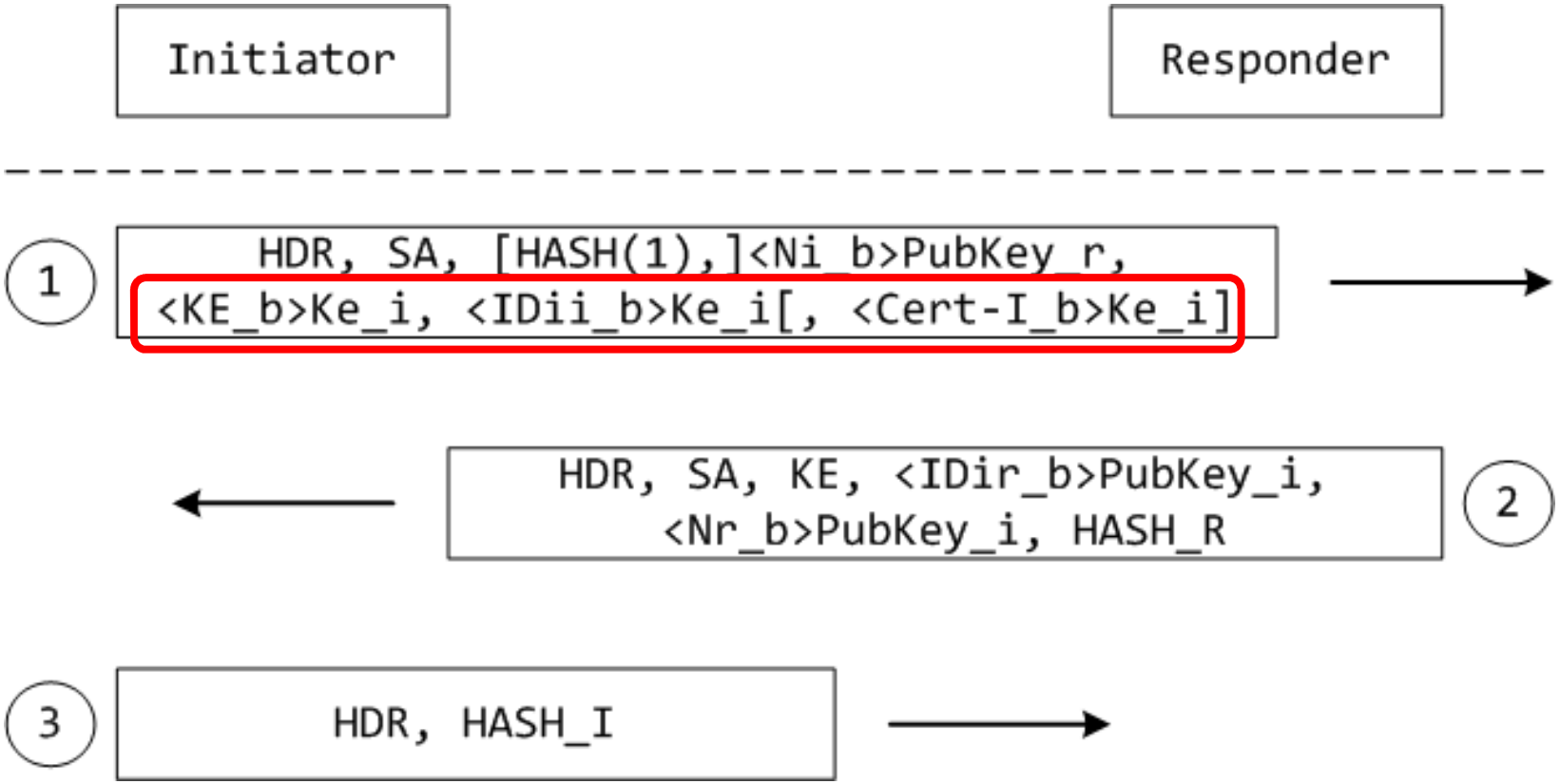
野蛮模式：使用公钥加密


Initiator
Responder
1
HDR, SA, [HASH(1),]KE,
<IDii_b>PubKey_r, <Ni_b>PubKey_r
HDR, SA, KE, <IDir_b>PubKey_i,
<Nr_b>PubKey_i, HASH_R
2
3
HDR, HASH_I

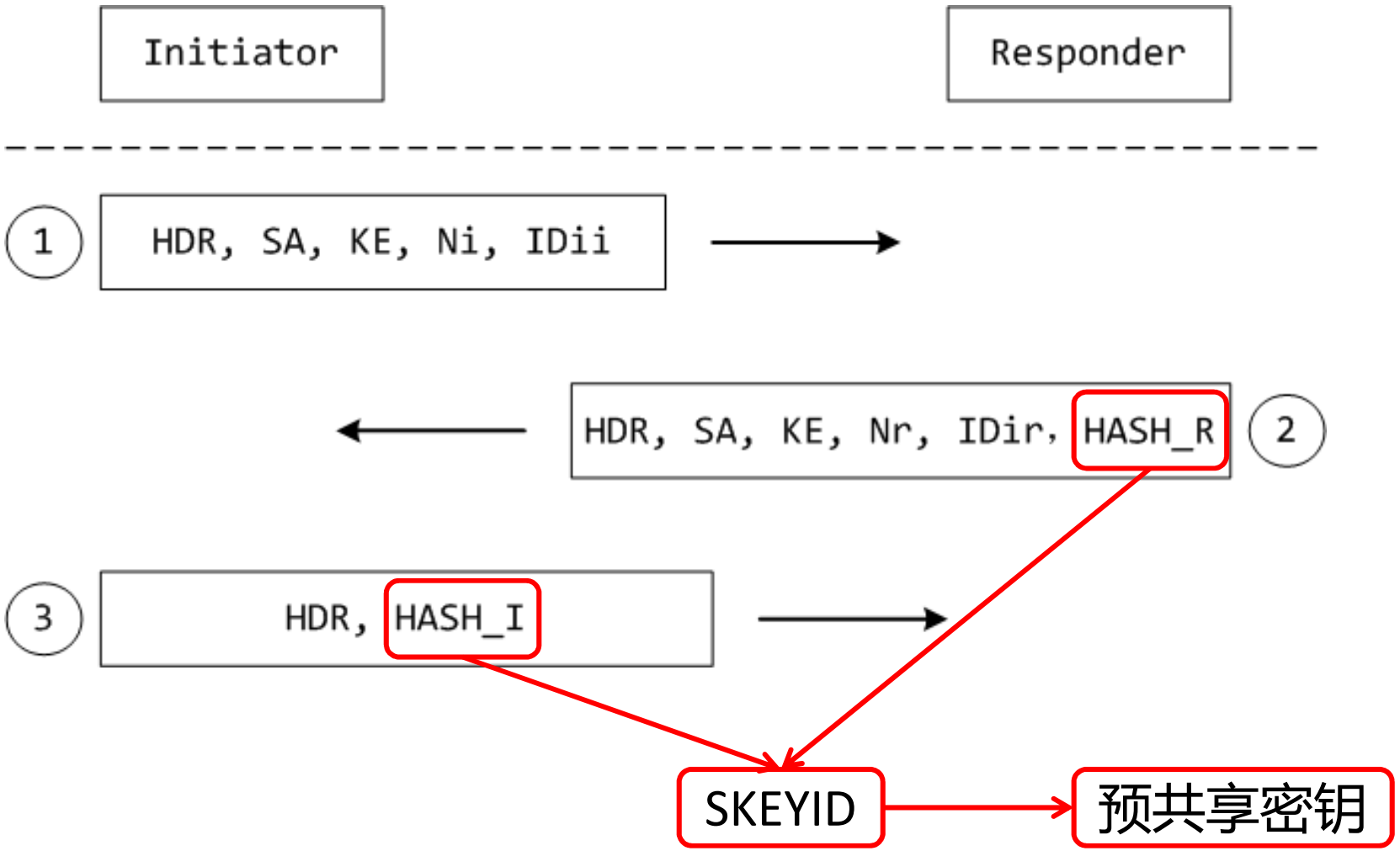
野蛮模式：使用公钥加密

- 第1个报文，HASH(1)的意义和前面主模式一样，用于向回应方说明所采用的公钥（当回应方有多个公钥的情况下）。
- 1、2报文中，ID和Nonce是用对方公钥加密的，要正确获得这些信息，接收方必须拥有对应的私钥。
- 通过2、3报文中HASH_I和HASH_R，通信双方可以验证对端身份。

野蛮模式：使用改进的公钥加密

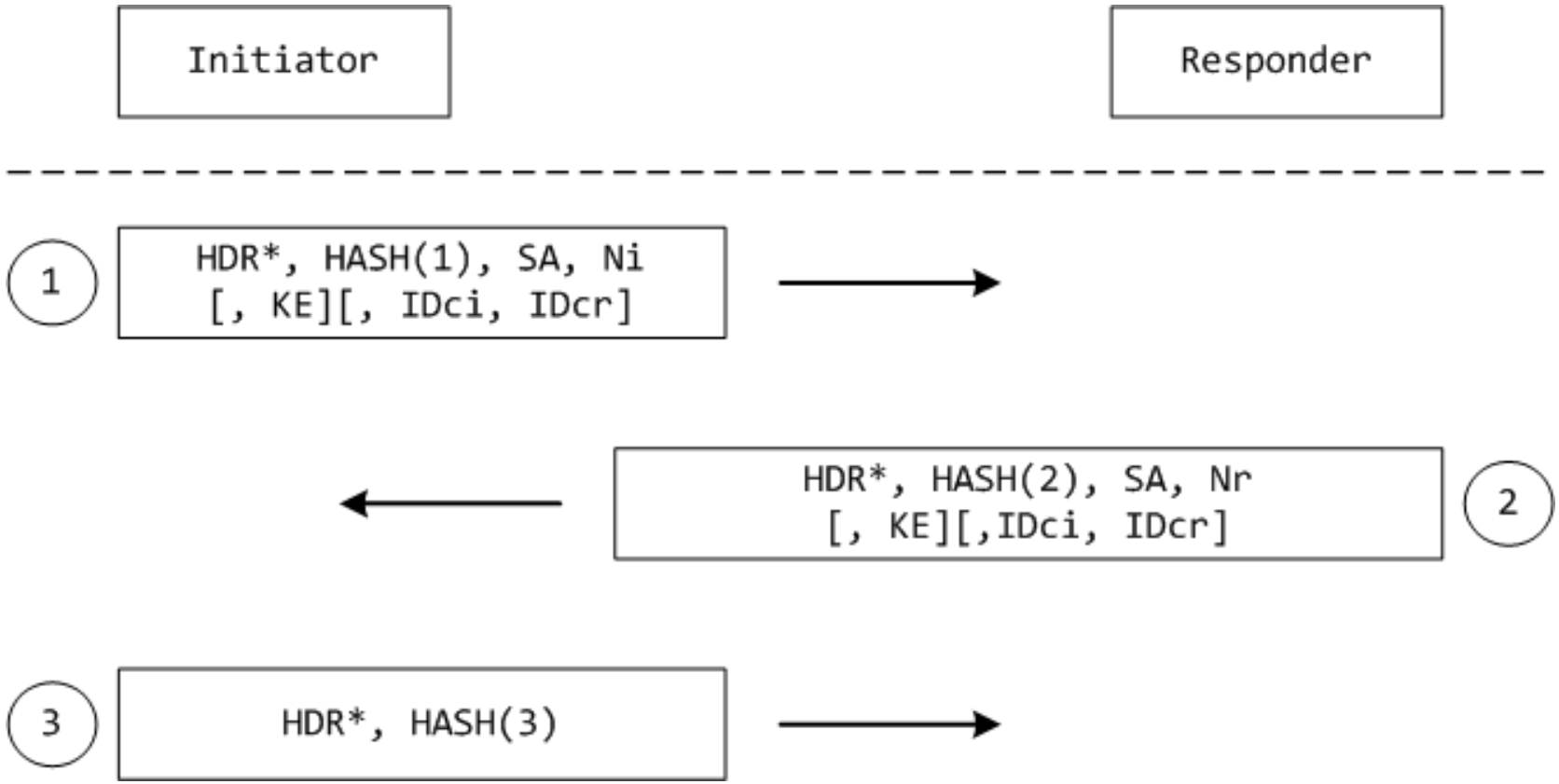


野蛮模式：使用预共享密钥



IKE第二阶段

IKE第二阶段：快速模式



- 3个hash值用于数据源认证和完整性校验

快速模式：HASH值的计算

HASH(1) =

prf(SKEYID_a, M-ID|SA|Ni[|KE][|IDci|IDcr])

HASH(2) =

prf(SKEYID_a,M-ID|Ni_b|SA|Nr [|KE][|IDci|IDcr])

HASH(3) =

prf(SKEYID_a, 0|M-ID|Ni_b|Nr_b)

SKEYID_a只有合法的通信对端知道，因此以此作为prf函数的输入之一，能够实现数据源发认证的功能，而其余输入则是报文内部的信息，能够实现完整性校验，起到防篡改的作用。

快速模式：密钥生成素材

- KE载荷是可选的，用于实现PFS。如果不需要PFS，就不需要交换KE载荷，新的密钥素材按照如下计算：

```
KEYMAT =
prf(SKEYID_d, protocol|SPI|Ni_b|Nr_b)
```

SKEYID_d:用于导出IPSec密钥的密钥素材

protocol、SPI：P载荷中包含的协议ID和SPI字段

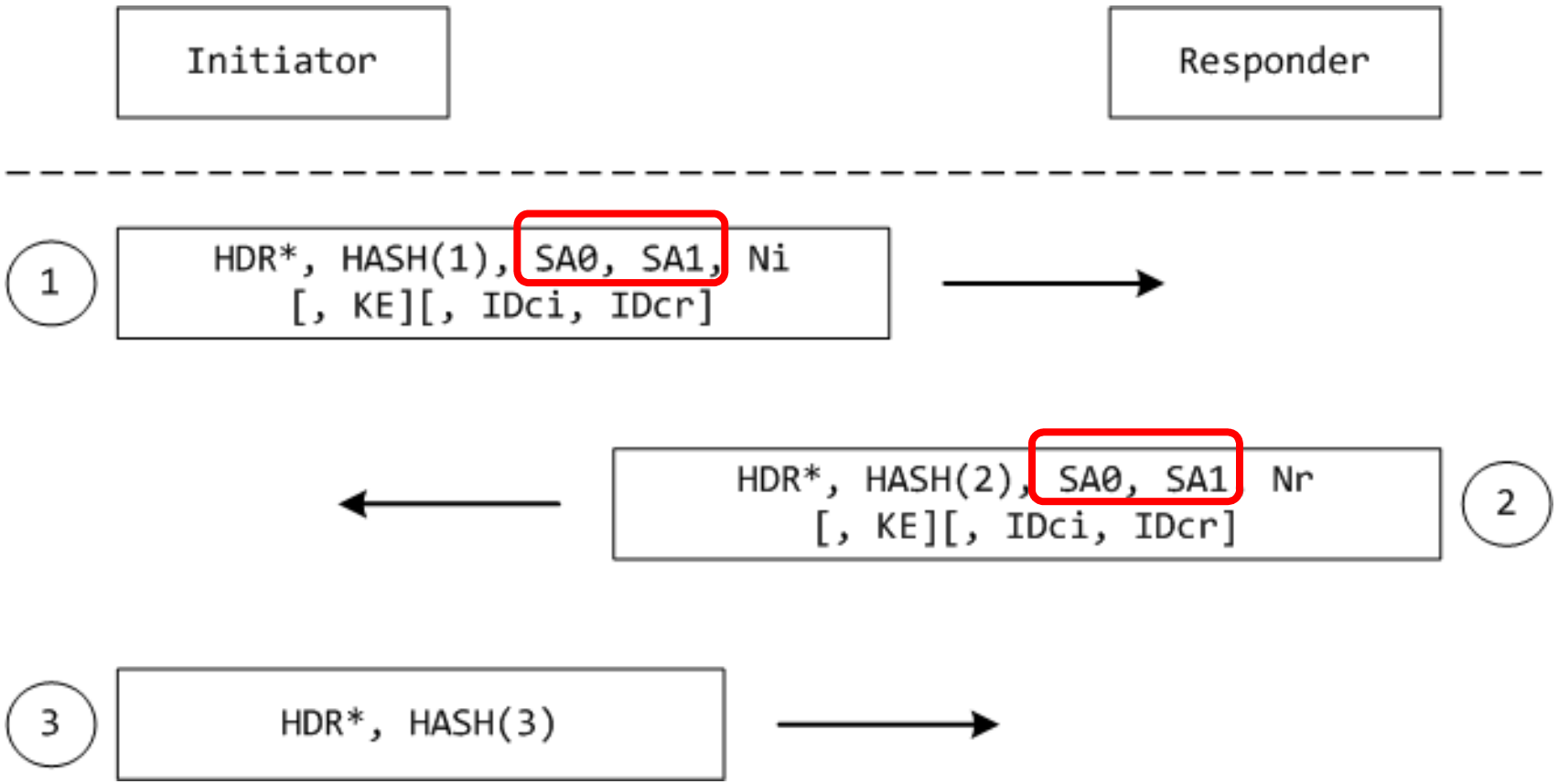
快速模式：密钥生成素材

- 如果需要PFS，就需要交换KE载荷，新的密钥素材按照如下计算：

```
KEYMAT =
prf(SKEYID_d, g(qm)^xy|protocol|SPI|Ni_b|Nr_b)
```

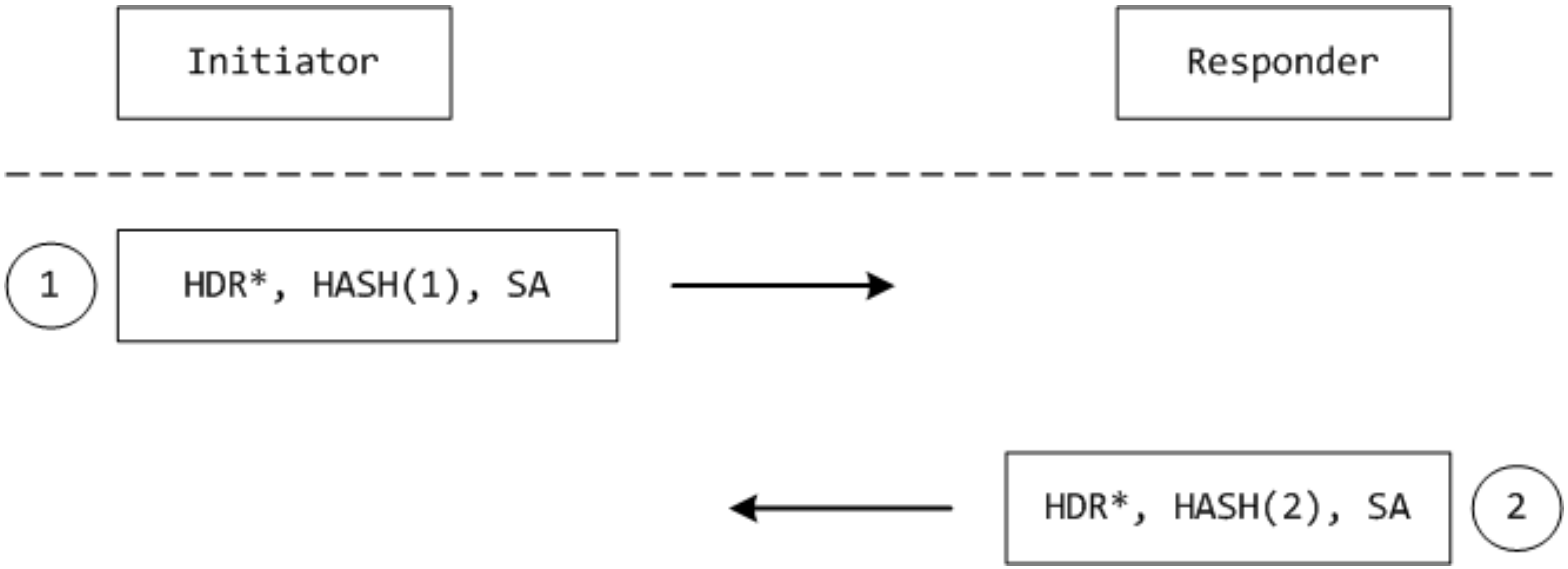
g(qm)^xy：快速模式下临时D-H交换的共享秘密

IKE第二阶段：快速模式（协商多个SA）



IKE第二阶段：新群模式

- 新群模式用于协商新的D-H群



HASH(1) = prf(SKEYID_a, M-ID|SA)

HASH(2) = prf(SKEYID_a, M-ID|SA)

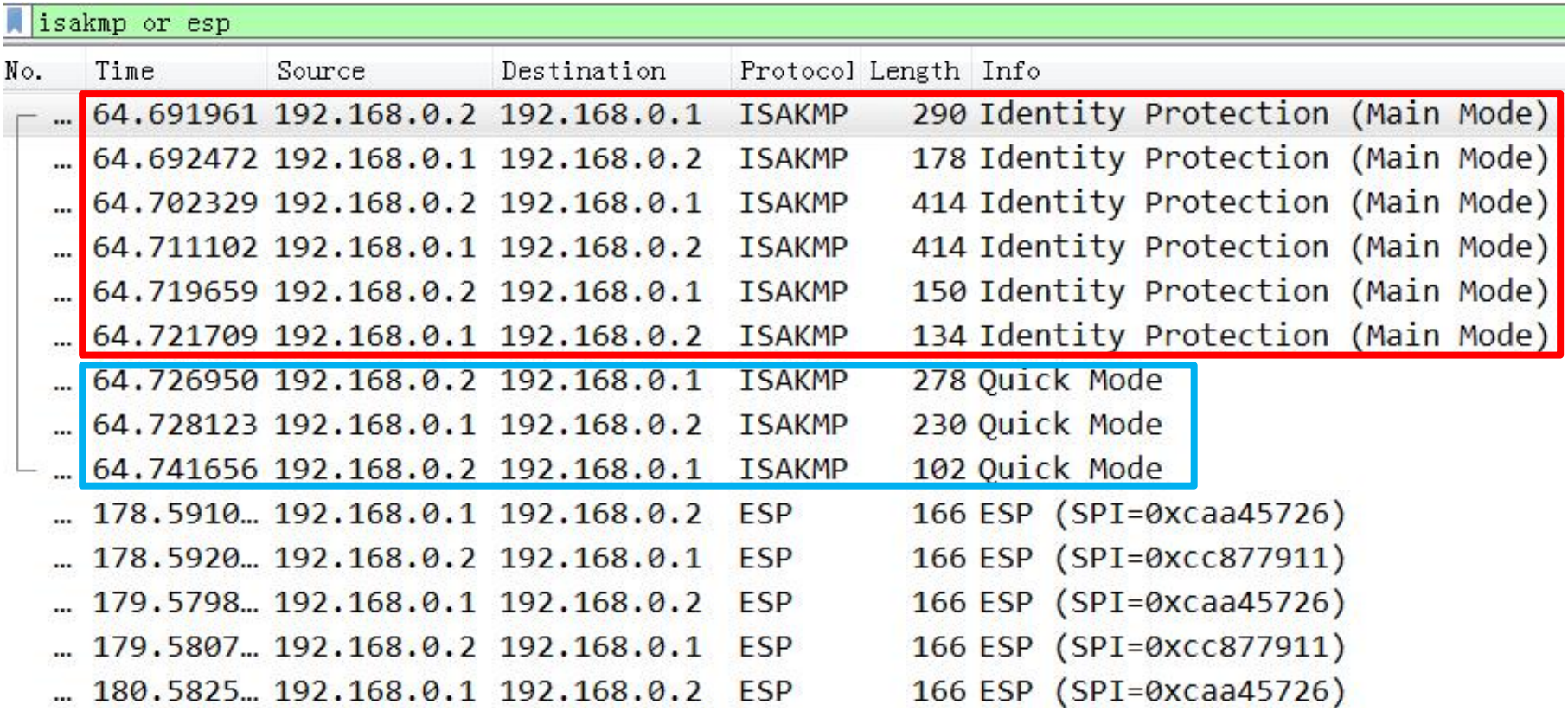
用于数据源发认证和消息完整性校验

IKE报文与载荷

- IKE报文沿用ISAKMP报文格式，报文由首部和数据区构成，数据区由载荷构成。
- 一次完整的IKE协商过程如何呢？

完整IKE协商示例

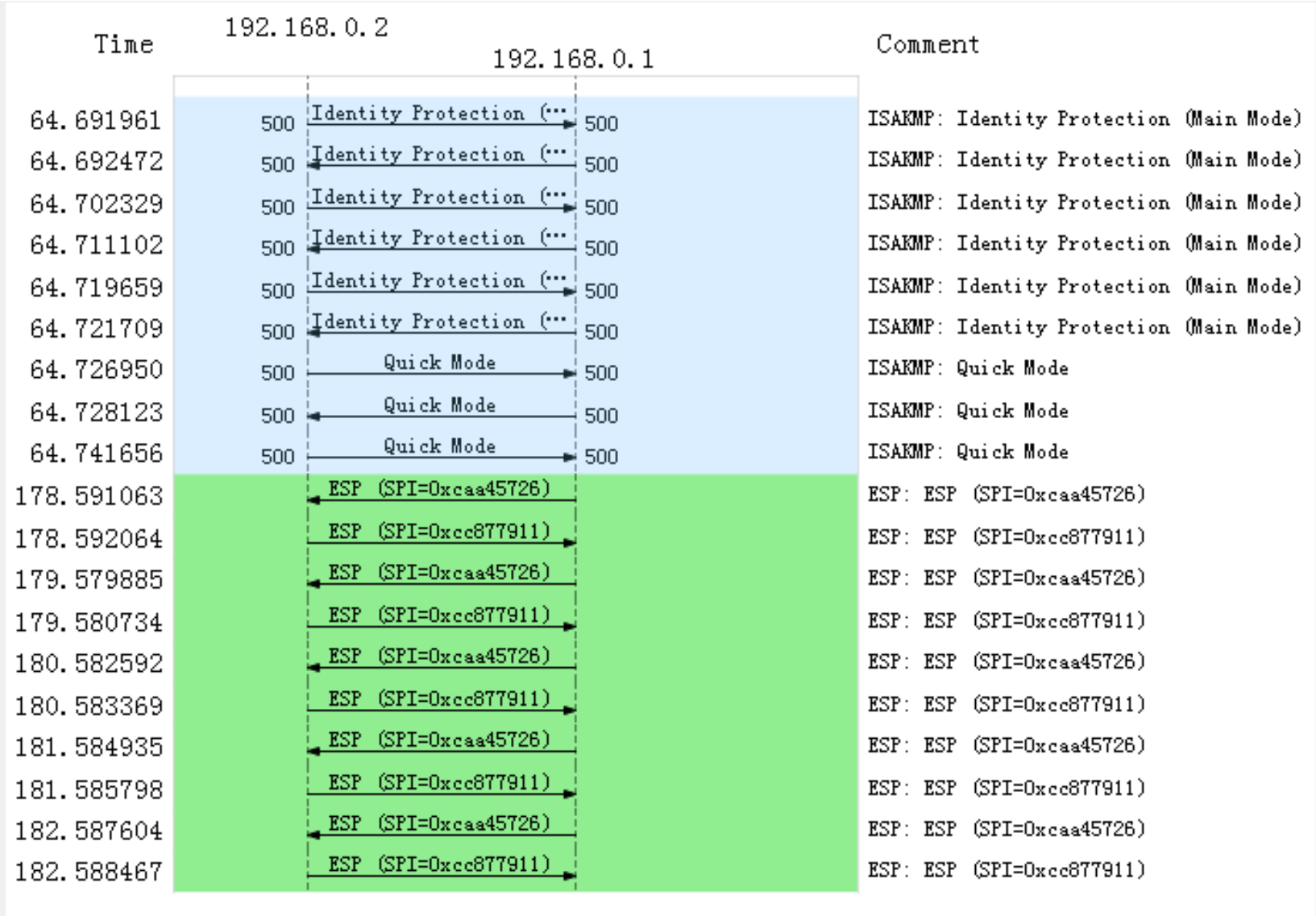
IKEv1实例分析



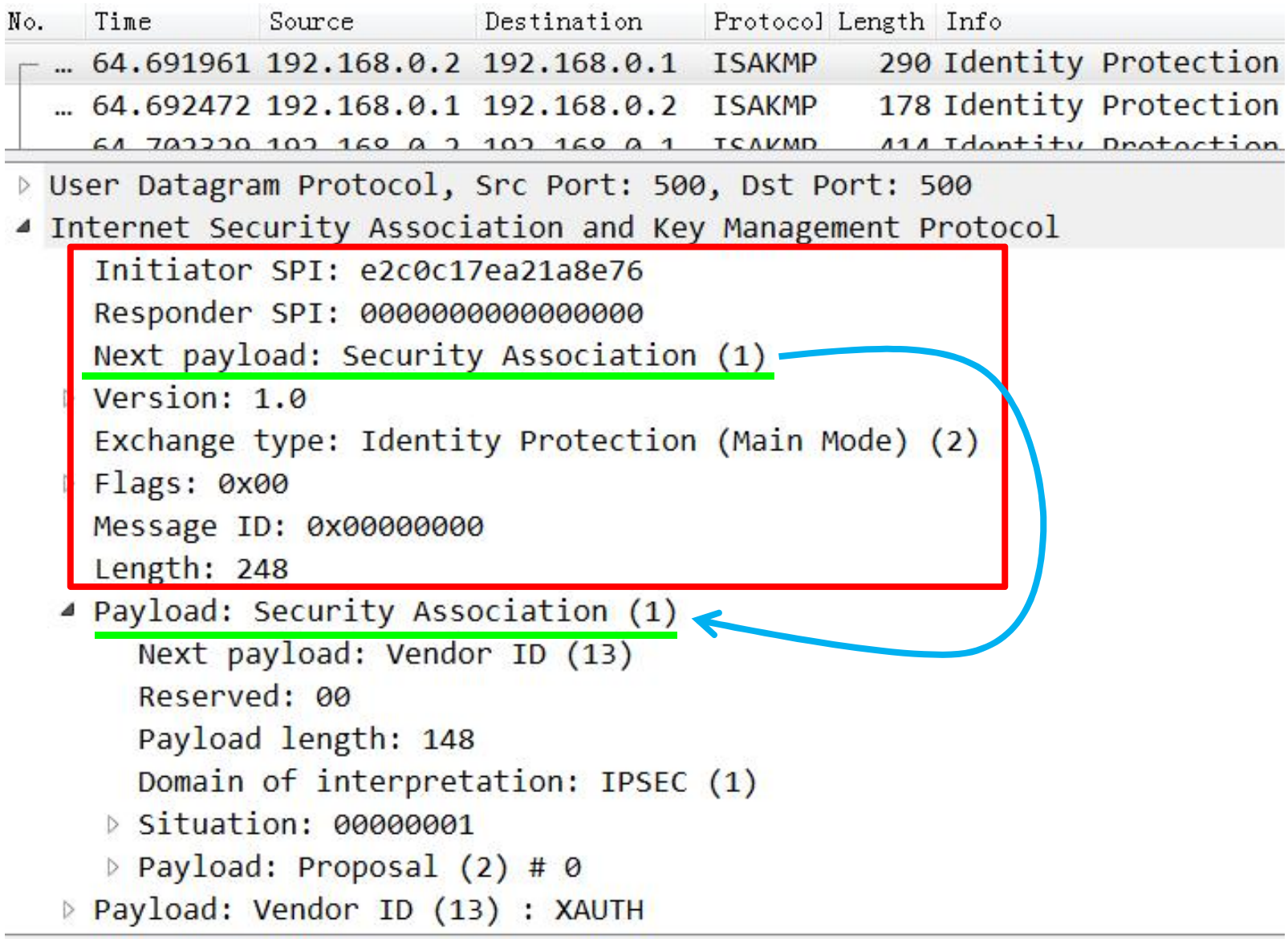
isakmp or esp

No.	Time	Source	Destination	Protocol	Length	Info
…	64.691961	192.168.0.2	192.168.0.1	ISAKMP	290	Identity Protection (Main Mode)
…	64.692472	192.168.0.1	192.168.0.2	ISAKMP	178	Identity Protection (Main Mode)
…	64.702329	192.168.0.2	192.168.0.1	ISAKMP	414	Identity Protection (Main Mode)
…	64.711102	192.168.0.1	192.168.0.2	ISAKMP	414	Identity Protection (Main Mode)
…	64.719659	192.168.0.2	192.168.0.1	ISAKMP	150	Identity Protection (Main Mode)
…	64.721709	192.168.0.1	192.168.0.2	ISAKMP	134	Identity Protection (Main Mode)
…	64.726950	192.168.0.2	192.168.0.1	ISAKMP	278	Quick Mode
…	64.728123	192.168.0.1	192.168.0.2	ISAKMP	230	Quick Mode
…	64.741656	192.168.0.2	192.168.0.1	ISAKMP	102	Quick Mode
…	178.5910…	192.168.0.1	192.168.0.2	ESP	166	ESP (SPI=0xcaa45726)
…	178.5920…	192.168.0.2	192.168.0.1	ESP	166	ESP (SPI=0xcc877911)
…	179.5798…	192.168.0.1	192.168.0.2	ESP	166	ESP (SPI=0xcaa45726)
…	179.5807…	192.168.0.2	192.168.0.1	ESP	166	ESP (SPI=0xcc877911)
…	180.5825…	192.168.0.1	192.168.0.2	ESP	166	ESP (SPI=0xcaa45726)

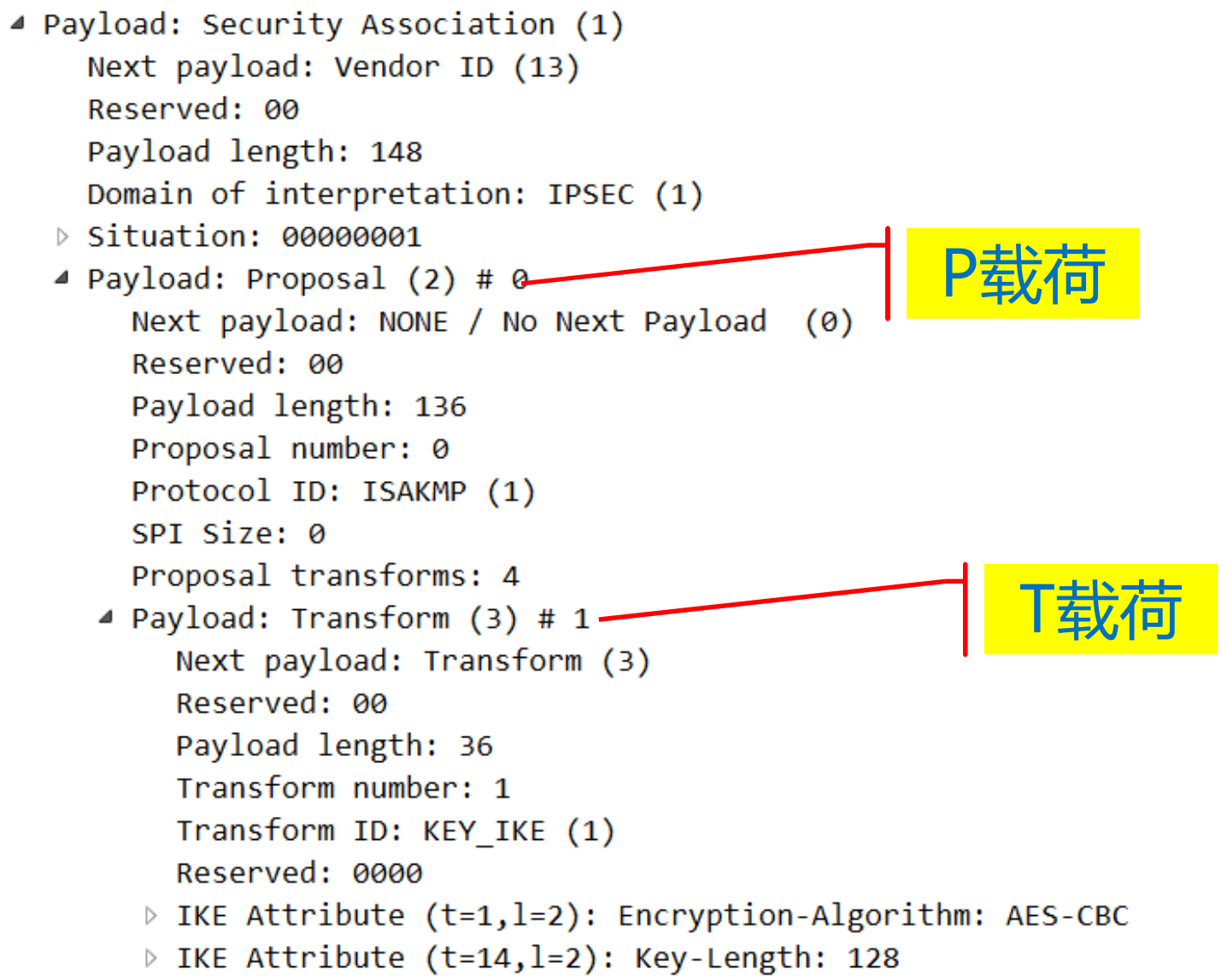
IKEv1实例：协议流程



IKEv1实例：消息1


No. Time Source Destination Protocol Length Info
… 64.691961 192.168.0.2 192.168.0.1 ISAKMP 290 Identity Protection
… 64.692472 192.168.0.1 192.168.0.2 ISAKMP 178 Identity Protection
User Datagram Protocol, Src Port: 500, Dst Port: 500
Internet Security Association and Key Management Protocol
Initiator SPI: e2c0c17ea21a8e76
Responder SPI: 0000000000000000
Next payload: Security Association (1)
Version: 1.0
Exchange type: Identity Protection (Main Mode) (2)
Flags: 0x00
Message ID: 0x00000000
Length: 248
Payload: Security Association (1)
Next payload: Vendor ID (13)
Reserved: 00
Payload length: 148
Domain of interpretation: IPSEC (1)
Situation: 00000001
Payload: Proposal (2) # 0
Payload: Vendor ID (13) : XAUTH

IKEv1实例：消息1


Payload: Security Association (1)
Next payload: Vendor ID (13)
Reserved: 00
Payload length: 148
Domain of interpretation: IPSEC (1)
Situation: 00000001
Payload: Proposal (2) # 0
Next payload: NONE / No Next Payload (0)
Reserved: 00
Payload length: 136
Proposal number: 0
Protocol ID: ISAKMP (1)
SPI Size: 0
Proposal transforms: 4
Payload: Transform (3) # 1
Next payload: Transform (3)
Reserved: 00
Payload length: 36
Transform number: 1
Transform ID: KEY_IKE (1)
Reserved: 0000
IKE Attribute (t=1,l=2): Encryption-Algorithm: AES-CBC
IKE Attribute (t=14,l=2): Key-Length: 128
P载荷
T载荷

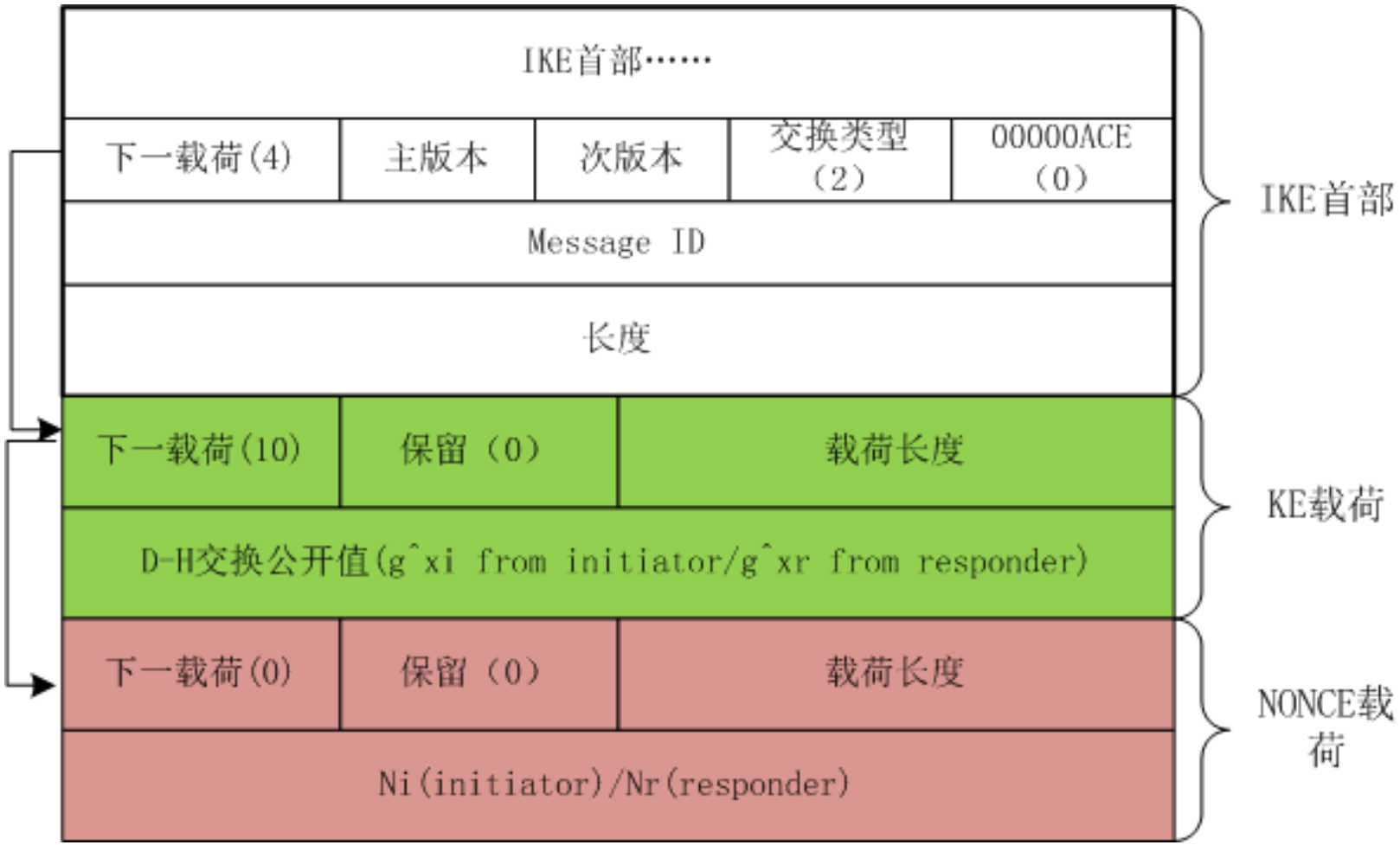
第2报文格式

- 第2报文格式与第1报文格式相同，只是回应方对建议的proposal进行了选择，也就是SA载荷中只有一个T载荷

IKEv1实例：消息2

```
▲ Payload: Proposal (2) # 0
    Next payload: NONE / No Next Payload  (0)
    Reserved: 00
    Payload length: 44
    Proposal number: 0
    Protocol ID: ISAKMP (1)
    SPI Size: 0
    Proposal transforms: 1
  ▲ Payload: Transform (3) # 1
      Next payload: NONE / No Next Payload  (0)
      Reserved: 00
      Payload length: 36
      Transform number: 1
      Transform ID: KEY_IKE (1)
      Reserved: 0000
    ▷ IKE Attribute (t=1,l=2): Encryption-Algorithm: AES-CBC
    ▷ IKE Attribute (t=14,l=2): Key-Length: 128
    ▷ IKE Attribute (t=2,l=2): Hash-Algorithm: SHA
    ▷ IKE Attribute (t=4,l=2): Group-Description: 2048 bit MODP group
    ▷ IKE Attribute (t=3,l=2): Authentication-Method: Pre-shared key
    ▷ IKE Attribute (t=11,l=2): Life-Type: Seconds
    ▷ IKE Attribute (t=12,l=2): Life-Duration: 3600
Payload: Vendor ID (13) : XAUTH
```

第3、4报文


IKE首部……
下一载荷(4)
主版本
次版本
交换类型（2）
00000ACE（0）
Message ID
长度
下一载荷(10)
保留（0）
载荷长度
D-H交换公开值(g^xi from initiator/g^xr from responder)
下一载荷(0)
保留（0）
载荷长度
Ni(initiator)/Nr(responder)
IKE首部
KE载荷
NONCE载荷

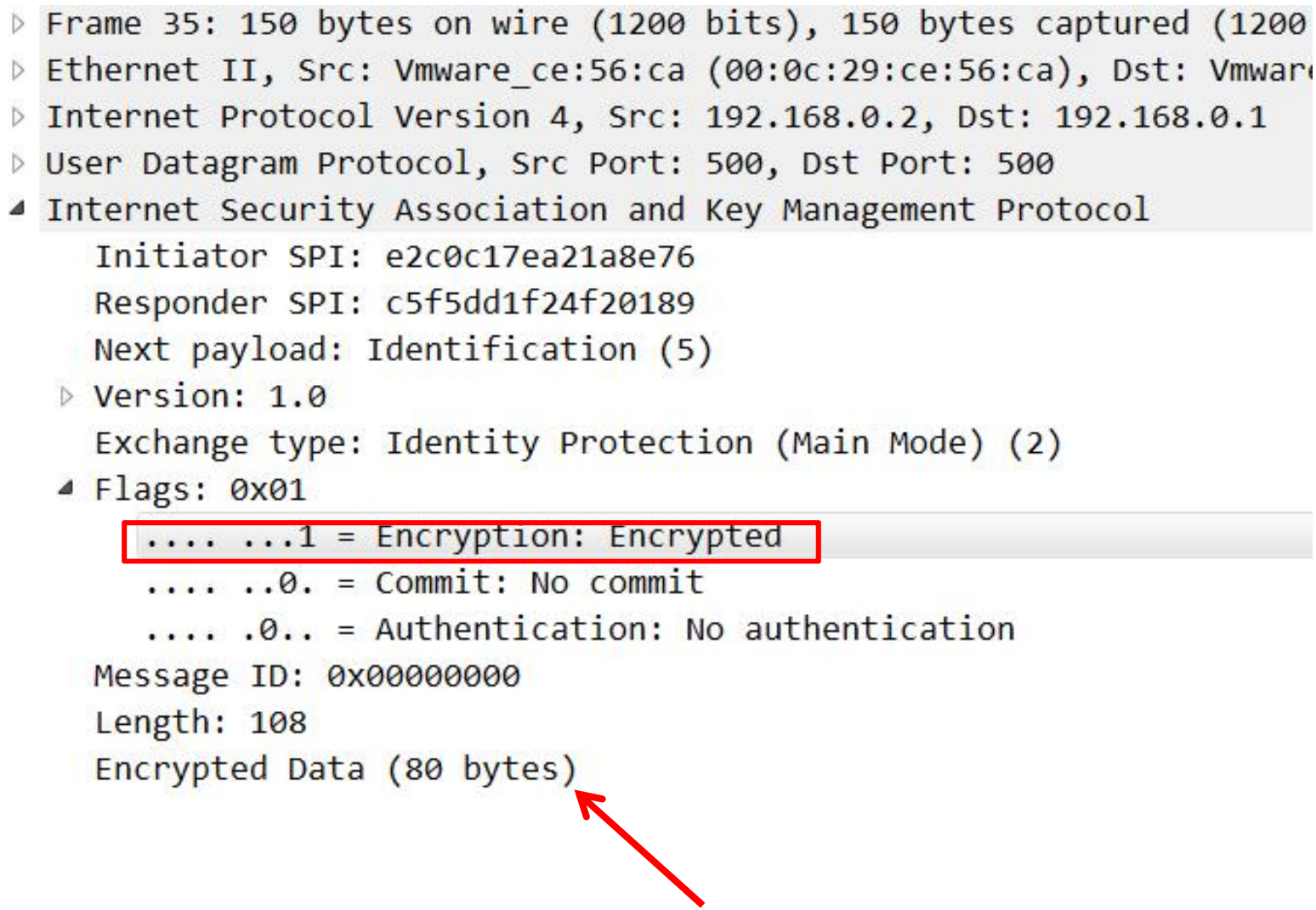
IKEv1实例：消息3

```
▲ Internet Security Association and Key Management Protocol
    Initiator SPI: e2c0c17ea21a8e76
    Responder SPI: c5f5dd1f24f20189
    Next payload: Key Exchange (4)
  ▷ Version: 1.0
    Exchange type: Identity Protection (Main Mode) (2)
  ▷ Flags: 0x00
    Message ID: 0x00000000
    Length: 372
  ▲ Payload: Key Exchange (4)
      Next payload: Nonce (10)
      Reserved: 00
      Payload length: 260
      Key Exchange Data: 7049e3f4af09d212d93a1ba2256fbb40e2eda40776544d39...
  ▲ Payload: Nonce (10)
      Next payload: NAT-D (RFC 3947) (20)
      Reserved: 00
      Payload length: 36
      Nonce DATA: b6af61710dbb6e7dc072f5f11211db408ee6695eb5df3943...
  ▷ Payload: NAT-D (RFC 3947) (20)
```

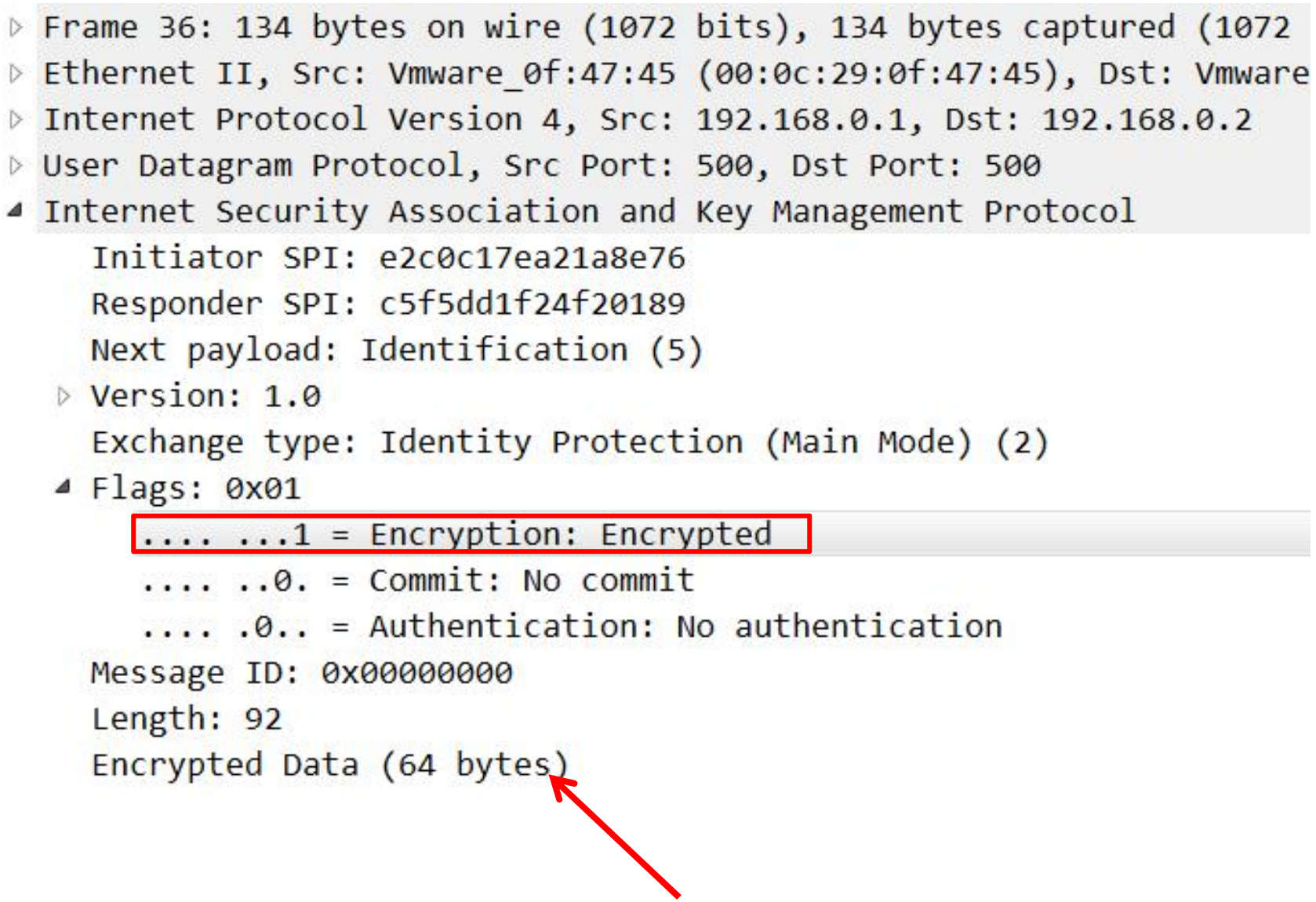
IKEv1实例：消息4

```
Internet Security Association and Key Management Protocol
    Initiator SPI: e2c0c17ea21a8e76
    Responder SPI: c5f5dd1f24f20189
    Next payload: Key Exchange (4)
  ▷ Version: 1.0
    Exchange type: Identity Protection (Main Mode) (2)
  ▷ Flags: 0x00
    Message ID: 0x00000000
    Length: 372
  ◢ Payload: Key Exchange (4)
        Next payload: Nonce (10)
        Reserved: 00
        Payload length: 260
        Key Exchange Data: 03deef458f66023f428892e34cc1e9767809def4ccdd1f18...
  ◢ Payload: Nonce (10)
        Next payload: NAT-D (RFC 3947) (20)
        Reserved: 00
        Payload length: 36
        Nonce DATA: 3f2fb964704ed803606ac2a0500dcc21963136daa049046e...
  ◢ Payload: NAT-D (RFC 3947) (20)
```

IKEv1实例：消息5


Frame 35: 150 bytes on wire (1200 bits), 150 bytes captured (1200
Ethernet II, Src: Vmware_ce:56:ca (00:0c:29:ce:56:ca), Dst: Vmwar
Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1
User Datagram Protocol, Src Port: 500, Dst Port: 500
Internet Security Association and Key Management Protocol
Initiator SPI: e2c0c17ea21a8e76
Responder SPI: c5f5dd1f24f20189
Next payload: Identification (5)
Version: 1.0
Exchange type: Identity Protection (Main Mode) (2)
Flags: 0x01
....1 = Encryption: Encrypted
.... ..0. = Commit: No commit
.... .0.. = Authentication: No authentication
Message ID: 0x00000000
Length: 108
Encrypted Data (80 bytes)

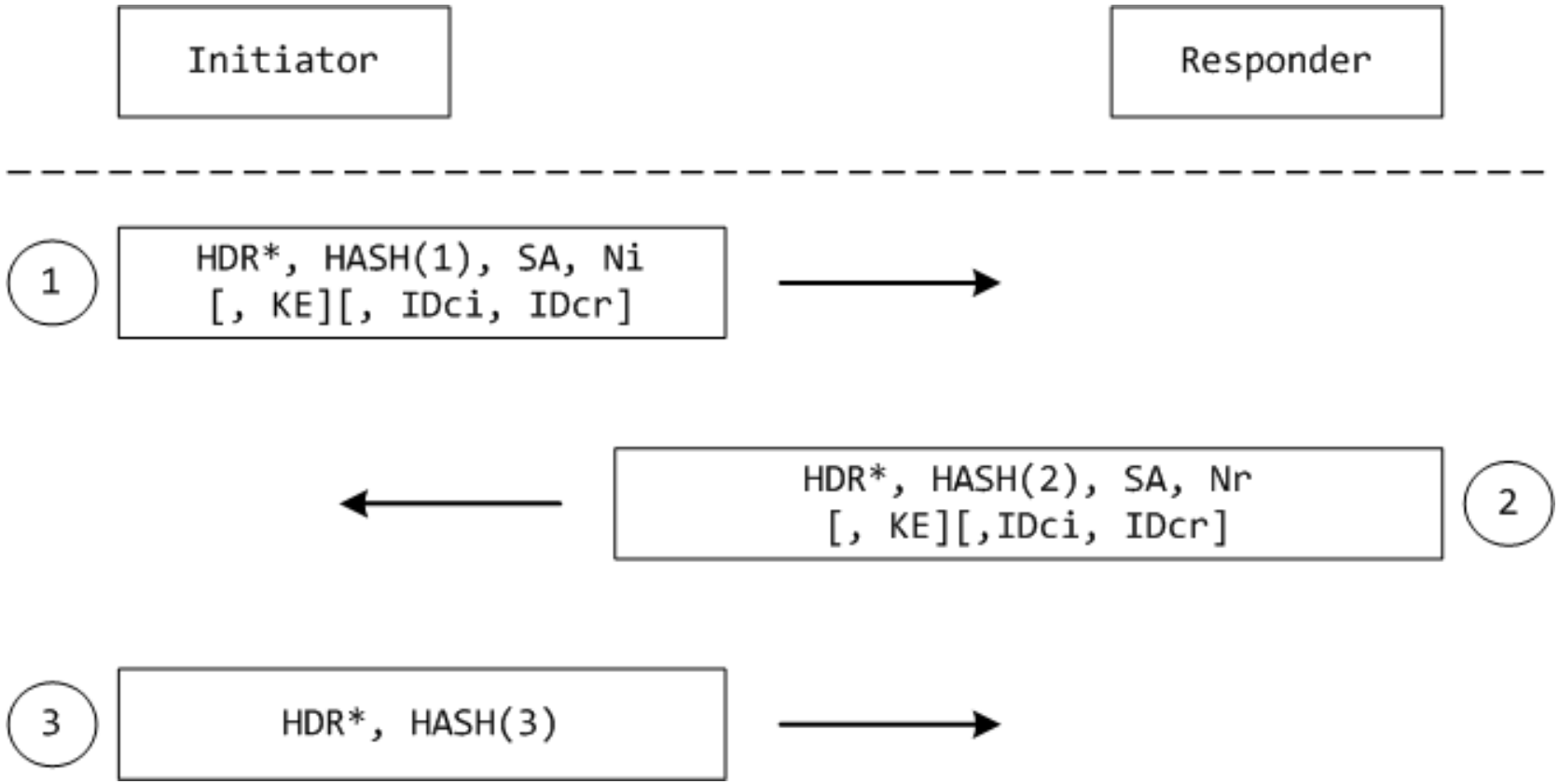
IKEv1实例：消息6


Frame 36: 134 bytes on wire (1072 bits), 134 bytes captured (1072
Ethernet II, Src: Vmware_0f:47:45 (00:0c:29:0f:47:45), Dst: Vmware
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2
User Datagram Protocol, Src Port: 500, Dst Port: 500
Internet Security Association and Key Management Protocol
Initiator SPI: e2c0c17ea21a8e76
Responder SPI: c5f5dd1f24f20189
Next payload: Identification (5)
Version: 1.0
Exchange type: Identity Protection (Main Mode) (2)
Flags: 0x01
....1 = Encryption: Encrypted
.... ..0. = Commit: No commit
.... .0.. = Authentication: No authentication
Message ID: 0x00000000
Length: 92
Encrypted Data (64 bytes)

IKE第二阶段：快速模式报文

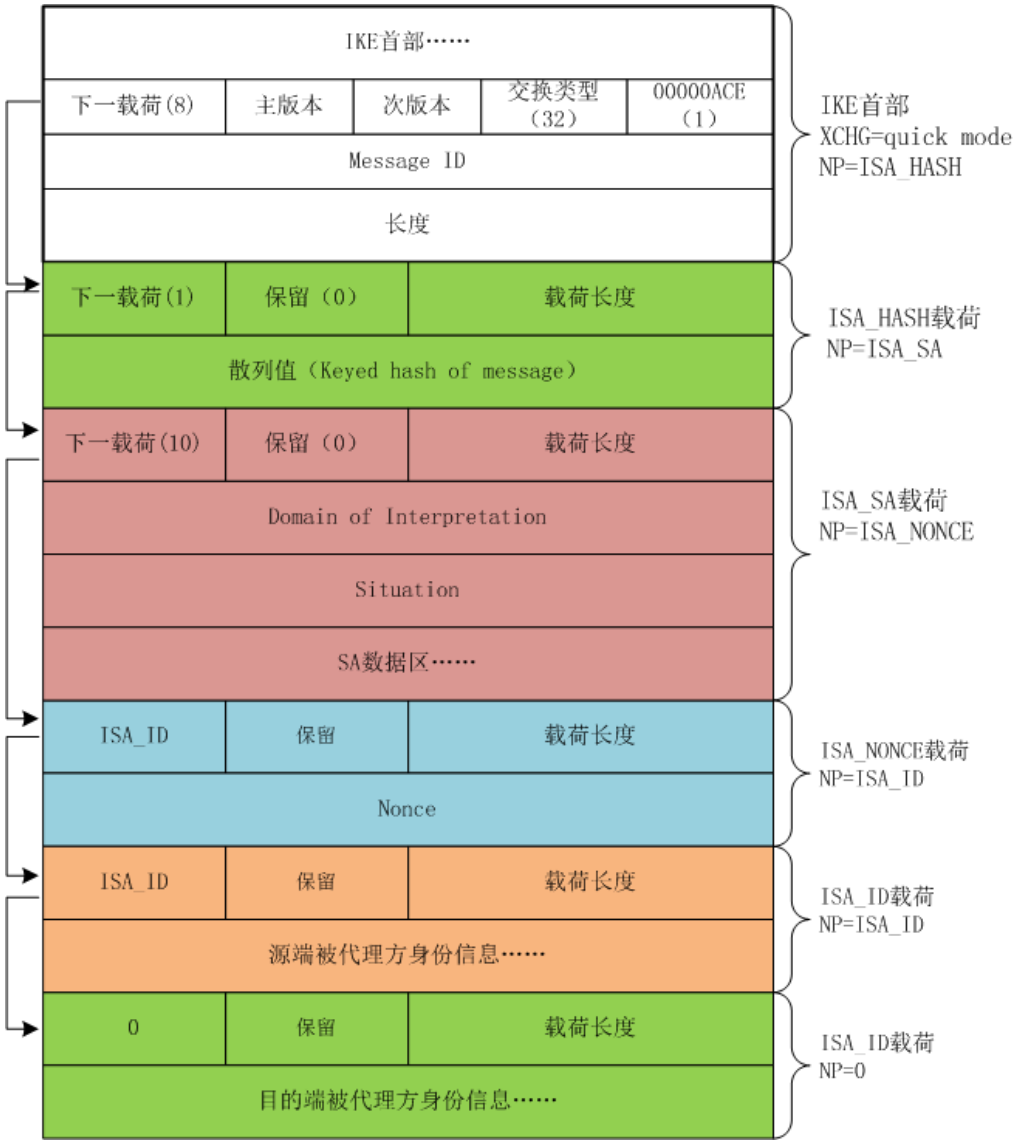
回顾一下快速模式
报文交换过程

IKE第二阶段：快速模式

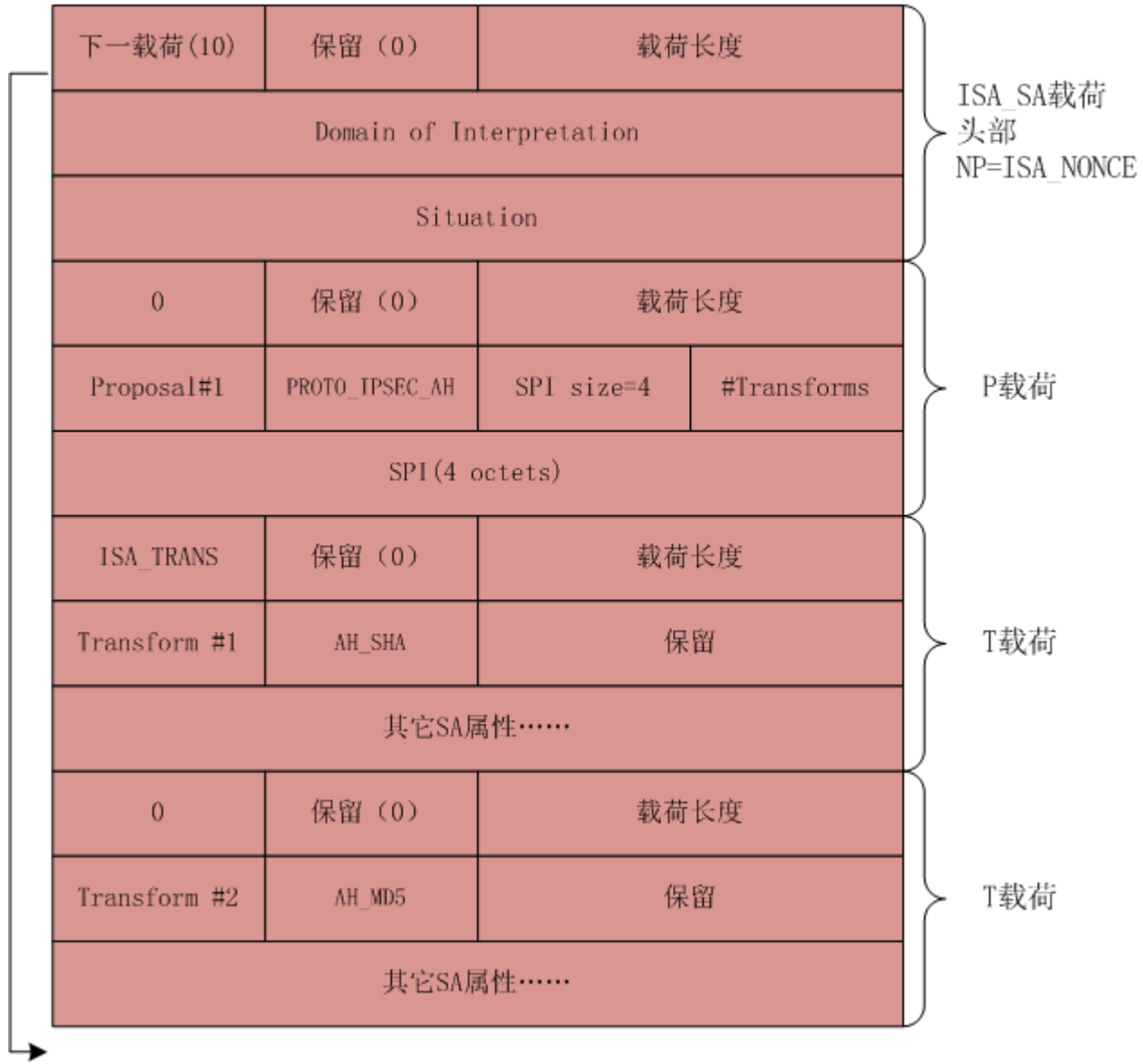


- 3个hash值用于数据源认证和完整性校验

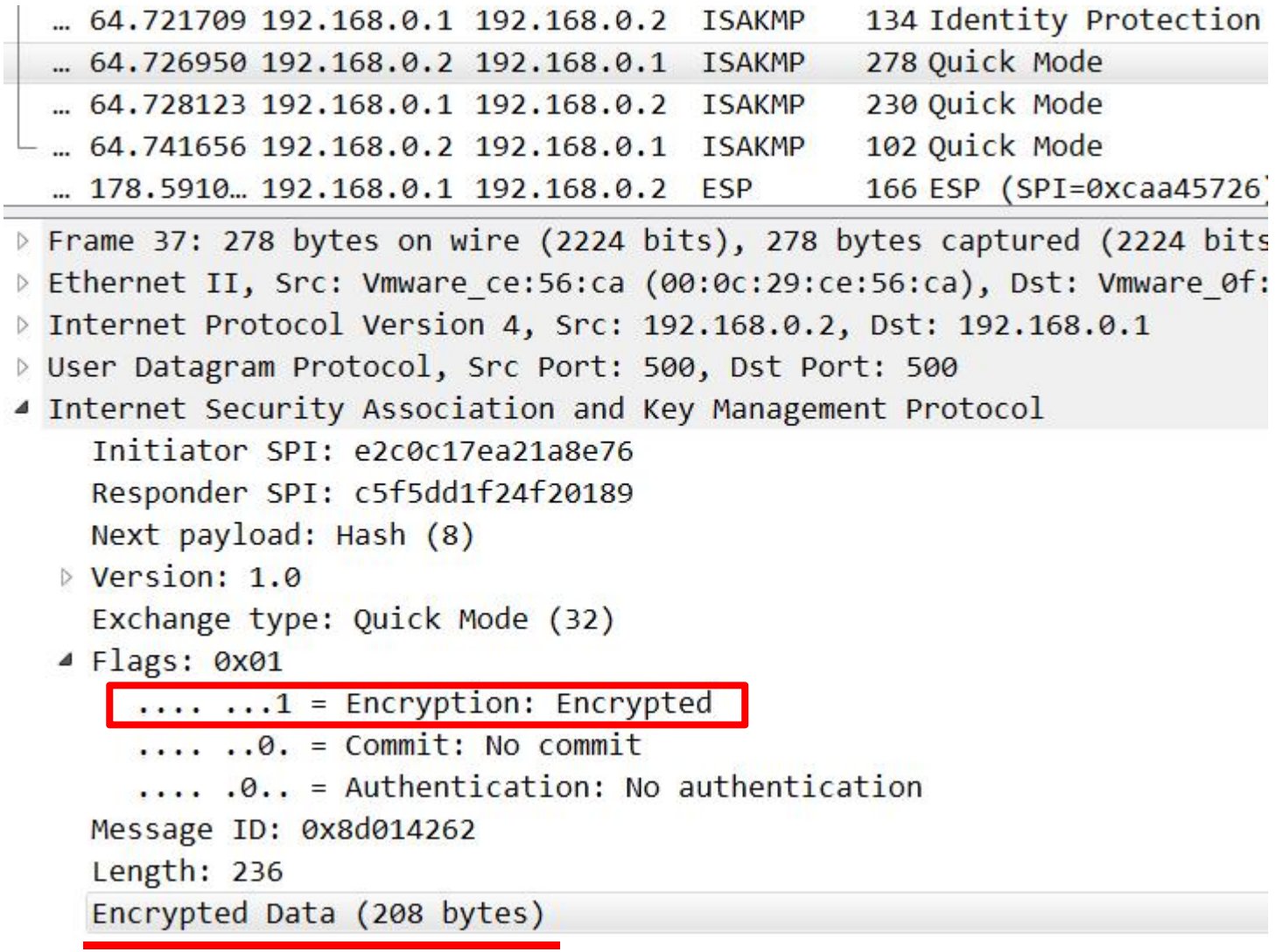
快速模式：第1报文格式


IKE首部……
下一载荷(8)
主版本
次版本
交换类型（32）
00000ACE（1）
Message ID
长度
IKE首部
XCHG=quick mode
NP=ISA_HASH
下一载荷(1)
保留（0）
载荷长度
散列值（Keyed hash of message）
ISA_HASH载荷
NP=ISA_SA
下一载荷(10)
保留（0）
载荷长度
Domain of Interpretation
Situation
SA数据区……
ISA_SA载荷
NP=ISA_NONCE
ISA_ID
保留
载荷长度
Nonce
ISA_NONCE载荷
NP=ISA_ID
ISA_ID
保留
载荷长度
源端被代理方身份信息……
ISA_ID载荷
NP=ISA_ID
0
保留
载荷长度
目的端被代理方身份信息……
ISA_ID载荷
NP=0

快速模式：第1、2报文SA载荷



IKEv1实例:第二阶段，消息1

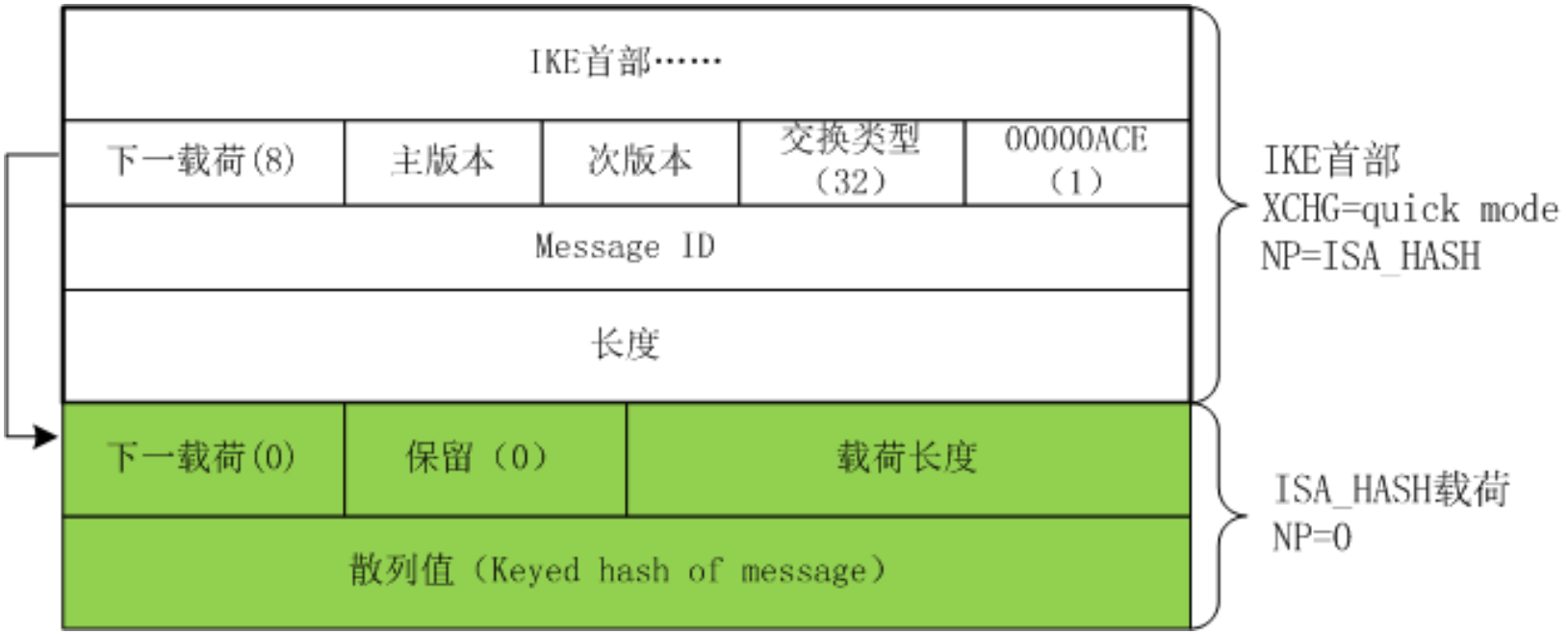


IKEv1实例:第二阶段，消息2

```
… 64.721709 192.168.0.1 192.168.0.2  ISAKMP    134 Identity Protection (M
… 64.726950 192.168.0.2 192.168.0.1  ISAKMP    278 Quick Mode
… 64.728123 192.168.0.1 192.168.0.2  ISAKMP    230 Quick Mode
… 64.741656 192.168.0.2 192.168.0.1  ISAKMP    102 Quick Mode
… 178.5910… 192.168.0.1 192.168.0.2  ESP       166 ESP (SPI=0xcaa45726)
```

```
▷ Frame 38: 230 bytes on wire (1840 bits), 230 bytes captured (1840 bits)
▷ Ethernet II, Src: Vmware_0f:47:45 (00:0c:29:0f:47:45), Dst: Vmware_ce:56
▷ Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2
▷ User Datagram Protocol, Src Port: 500, Dst Port: 500
◢ Internet Security Association and Key Management Protocol
    Initiator SPI: e2c0c17ea21a8e76
    Responder SPI: c5f5dd1f24f20189
    Next payload: Hash (8)
  ▷ Version: 1.0
    Exchange type: Quick Mode (32)
  ◢ Flags: 0x01
      .... ...1 = Encryption: Encrypted
      .... ..0. = Commit: No commit
      .... .0.. = Authentication: No authentication
    Message ID: 0x8d014262
    Length: 188
    Encrypted Data (160 bytes)
```

快速模式：第3个报文


IKE首部……
下一载荷(8)
主版本
次版本
交换类型
(32)
00000ACE
(1)
Message ID
长度
下一载荷(0)
保留（0）
载荷长度
散列值（Keyed hash of message）
IKE首部
XCHG=quick mode
NP=ISA_HASH
ISA_HASH载荷
NP=0

IKEv1实例:第二阶段，消息3

```
… 64.721709 192.168.0.1 192.168.0.2 ISAKMP    134 Identity Protectio
… 64.726950 192.168.0.2 192.168.0.1 ISAKMP    278 Quick Mode
… 64.728123 192.168.0.1 192.168.0.2 ISAKMP    230 Quick Mode
… 64.741656 192.168.0.2 192.168.0.1 ISAKMP    102 Quick Mode
… 178.5910… 192.168.0.1 192.168.0.2 ESP       166 ESP (SPI=0xcaa4572
```

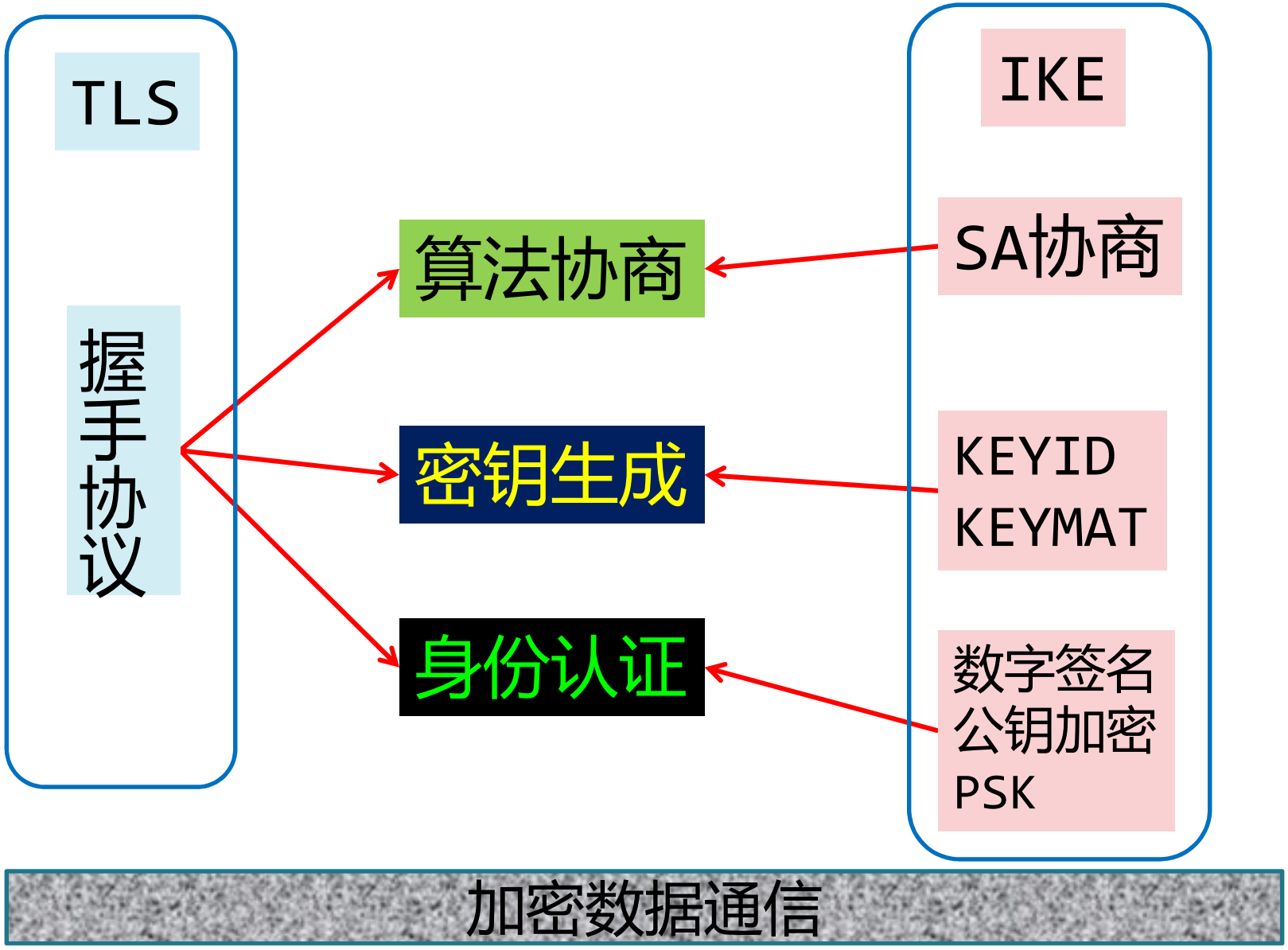
```
▷ Frame 39: 102 bytes on wire (816 bits), 102 bytes captured (816 bits)
▷ Ethernet II, Src: Vmware_ce:56:ca (00:0c:29:ce:56:ca), Dst: Vmware_0
▷ Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1
▷ User Datagram Protocol, Src Port: 500, Dst Port: 500
◢ Internet Security Association and Key Management Protocol
    Initiator SPI: e2c0c17ea21a8e76
    Responder SPI: c5f5dd1f24f20189
    Next payload: Hash (8)
  ▷ Version: 1.0
    Exchange type: Quick Mode (32)
  ◢ Flags: 0x01
      .... ...1 = Encryption: Encrypted
      .... ..0. = Commit: No commit
      .... .0.. = Authentication: No authentication
    Message ID: 0x8d014262
    Length: 60
    Encrypted Data (32 bytes)
```

IKEv1的不足

- 协商IPSec SA所需消息开销较大
  - **主模式**协商一对IPSec SA，需要6（协商IKE SA）+3（协商IPSec SA）= 9条消息
  - **野蛮模式**协商一对IPSec SA，需要3（协商IKE SA）+3（协商IPSec SA）= 6条消息

IKEv2简介

TLS vs IKE


TLS
握手协议
算法协商
密钥生成
身份认证
IKE
SA协商
KEYID
KEYMAT
数字签名
公钥加密
PSK
加密数据通信

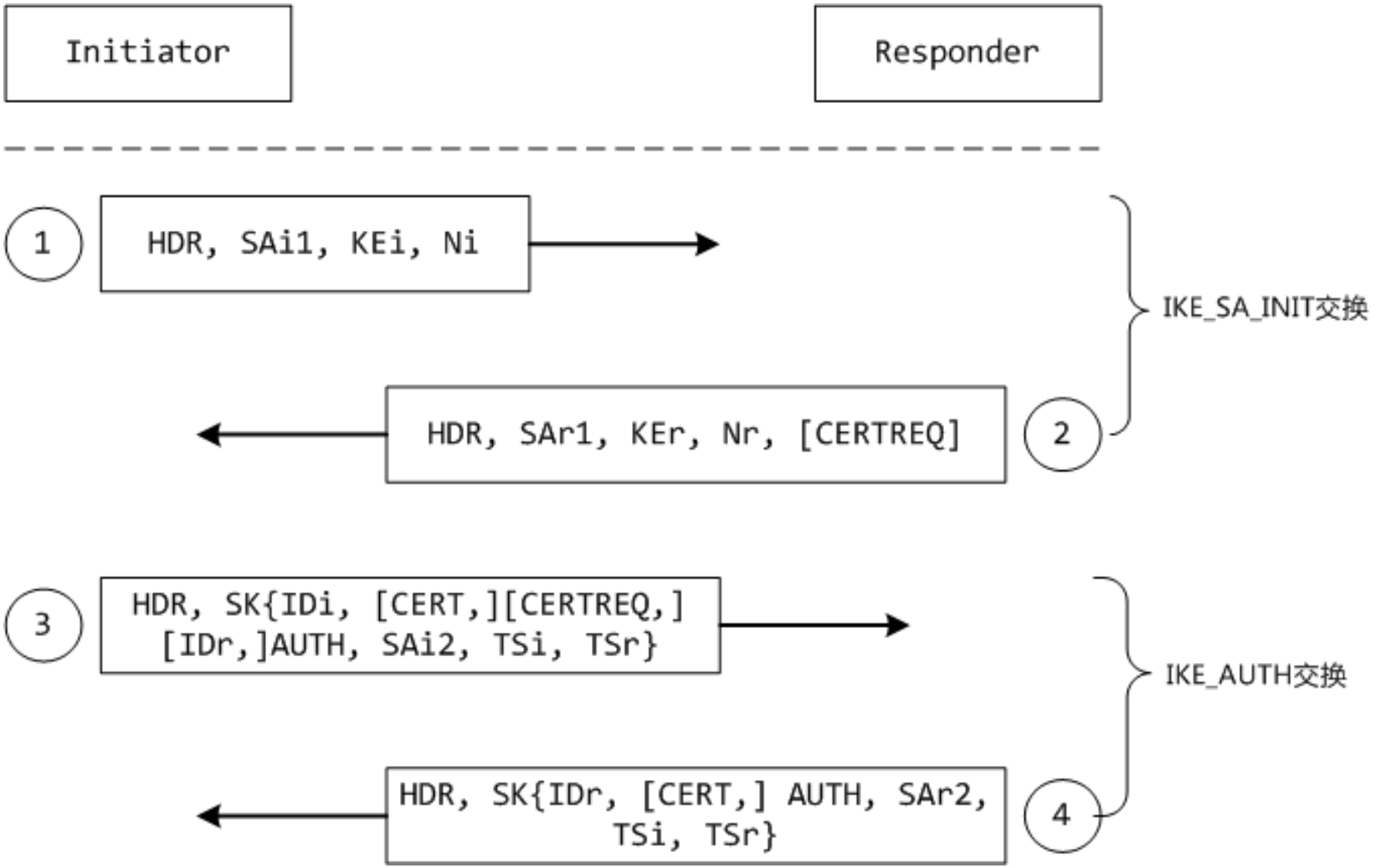
IKEv1

- 第一阶段，协商获得IKE SA
  - 主模式（main mode）
    - 对应ISAKMP：Identity Protect Exchange
  - 野蛮模式（aggressive mode）
    - 对应ISAKMP：Aggressive Exchange
- 第二阶段，协商获得IPSec SA
  - 快速模式

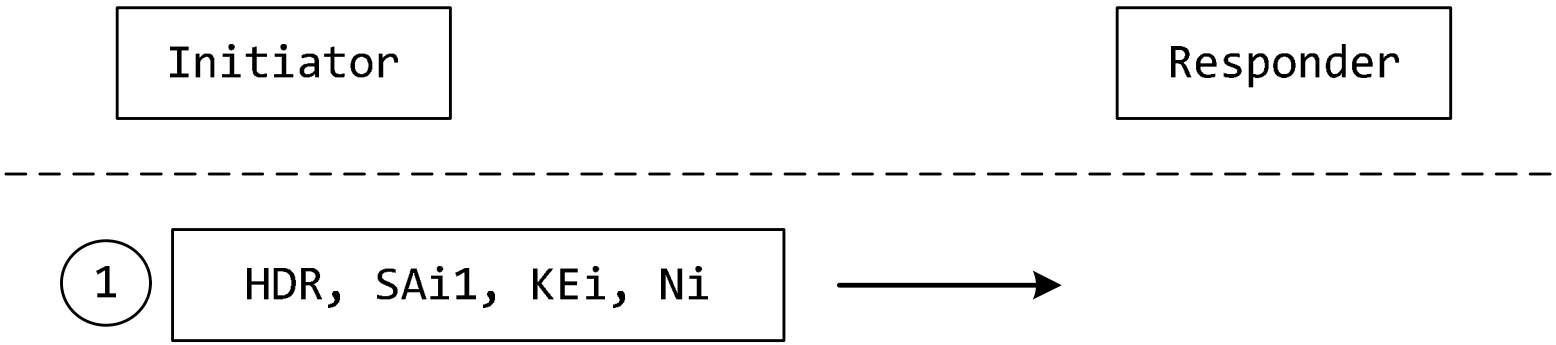
IKEv1的不足

- 协商IPSec SA所需消息开销较大
  - **主模式**协商一对IPSec SA，需要6（协商IKE SA）+3（协商IPSec SA）= 9条消息
  - **野蛮模式**协商一对IPSec SA，需要3（协商IKE SA）+3（协商IPSec SA）= 6条消息

IKEv2消息交互


Initiator
Responder
1
HDR, SAi1, KEi, Ni
IKE_SA_INIT交换
HDR, SAr1, KEr, Nr, [CERTREQ]
2
3
HDR, SK{IDi, [CERT,][CERTREQ,]
[IDr,]AUTH, SAi2, TSi, TSr}
IKE_AUTH交换
HDR, SK{IDr, [CERT,] AUTH, SAr2,
TSi, TSr}
4

IKE_SA_INIT：第一条消息

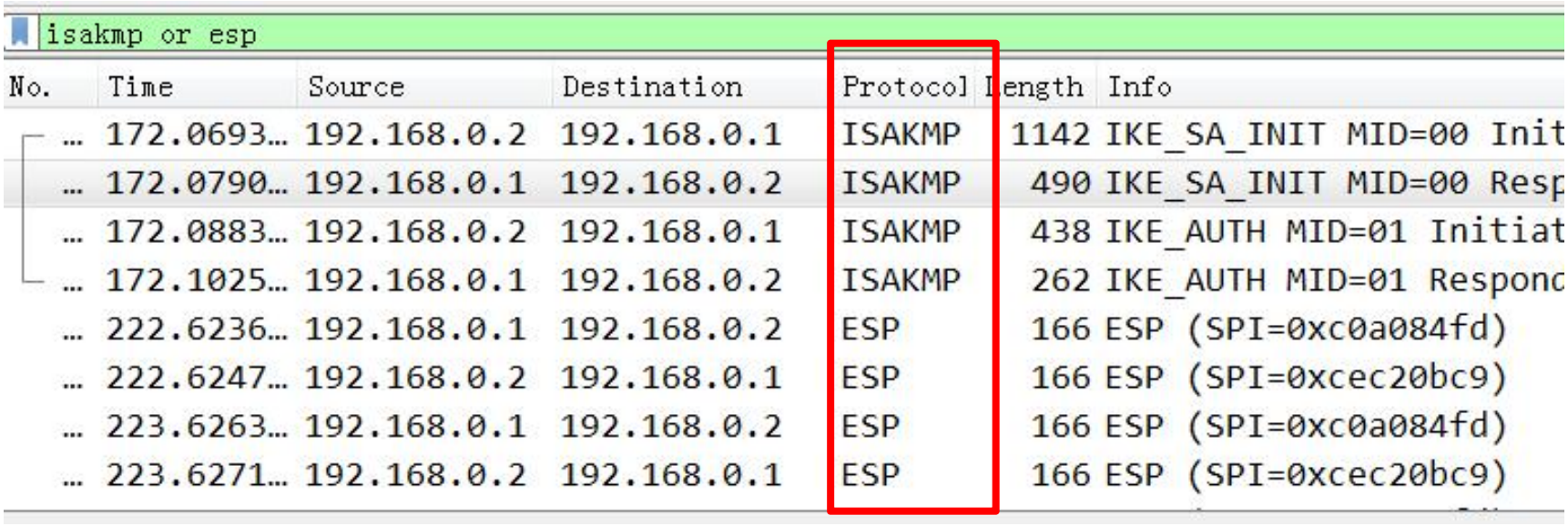


- HDR包含SPIs、版本号、flags；
- SAi1载荷说明发起方为保护IKE通信所支持的密码算法；
- KEi载荷发送发起方的DH值；
- Ni是发起方的Nonce。

IKE_SA_INIT：第2条消息

回应方从发起方提供的选项中选择一个密码套件，并用SAr1载荷来表明自己的选择，用KEr载荷来完成DH交换，用Nr载荷来发送回应方的Nonce。

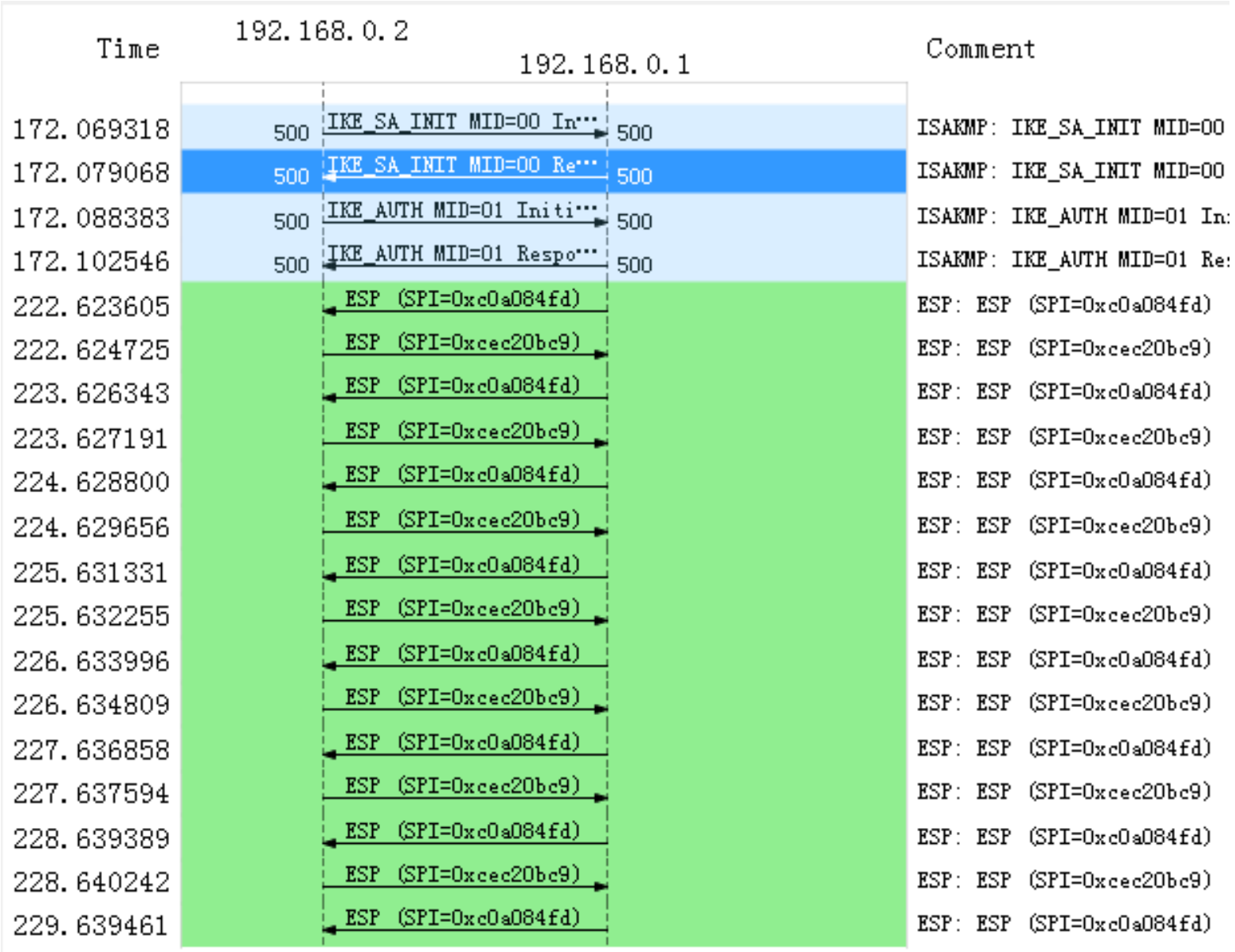
IKEv2实例



isakmp or esp

No.	Time	Source	Destination	Protocol	Length	Info
…	172.0693…	192.168.0.2	192.168.0.1	ISAKMP	1142	IKE_SA_INIT MID=00 Init
…	172.0790…	192.168.0.1	192.168.0.2	ISAKMP	490	IKE_SA_INIT MID=00 Resp
…	172.0883…	192.168.0.2	192.168.0.1	ISAKMP	438	IKE_AUTH MID=01 Initiat
…	172.1025…	192.168.0.1	192.168.0.2	ISAKMP	262	IKE_AUTH MID=01 Respond
…	222.6236…	192.168.0.1	192.168.0.2	ESP	166	ESP (SPI=0xc0a084fd)
…	222.6247…	192.168.0.2	192.168.0.1	ESP	166	ESP (SPI=0xcec20bc9)
…	223.6263…	192.168.0.1	192.168.0.2	ESP	166	ESP (SPI=0xc0a084fd)
…	223.6271…	192.168.0.2	192.168.0.1	ESP	166	ESP (SPI=0xcec20bc9)

IKEv2实例


Time
192.168.0.2
192.168.0.1
Comment
172.069318
500
IKE_SA_INIT MID=00 In…
500
ISAKMP: IKE_SA_INIT MID=00
172.079068
500
IKE_SA_INIT MID=00 Re…
500
ISAKMP: IKE_SA_INIT MID=00
172.088383
500
IKE_AUTH MID=01 Initi…
500
ISAKMP: IKE_AUTH MID=01 In
172.102546
500
IKE_AUTH MID=01 Respo…
500
ISAKMP: IKE_AUTH MID=01 Re
222.623605
ESP (SPI=0xc0a084fd)
ESP: ESP (SPI=0xc0a084fd)
222.624725
ESP (SPI=0xcec20bc9)
ESP: ESP (SPI=0xcec20bc9)
223.626343
ESP (SPI=0xc0a084fd)
ESP: ESP (SPI=0xc0a084fd)
223.627191
ESP (SPI=0xcec20bc9)
ESP: ESP (SPI=0xcec20bc9)
224.628800
ESP (SPI=0xc0a084fd)
ESP: ESP (SPI=0xc0a084fd)
224.629656
ESP (SPI=0xcec20bc9)
ESP: ESP (SPI=0xcec20bc9)
225.631331
ESP (SPI=0xc0a084fd)
ESP: ESP (SPI=0xc0a084fd)
225.632255
ESP (SPI=0xcec20bc9)
ESP: ESP (SPI=0xcec20bc9)
226.633996
ESP (SPI=0xc0a084fd)
ESP: ESP (SPI=0xc0a084fd)
226.634809
ESP (SPI=0xcec20bc9)
ESP: ESP (SPI=0xcec20bc9)
227.636858
ESP (SPI=0xc0a084fd)
ESP: ESP (SPI=0xc0a084fd)
227.637594
ESP (SPI=0xcec20bc9)
ESP: ESP (SPI=0xcec20bc9)
228.639389
ESP (SPI=0xc0a084fd)
ESP: ESP (SPI=0xc0a084fd)
228.640242
ESP (SPI=0xcec20bc9)
ESP: ESP (SPI=0xcec20bc9)
229.639461
ESP (SPI=0xc0a084fd)
ESP: ESP (SPI=0xc0a084fd)

IKEv2实例：消息1

```
▲ Internet Security Association and Key Management Protocol
    Initiator SPI: 0dda36cf9a109c86
    Responder SPI: 0000000000000000
    Next payload: Security Association (33)
  ▷ Version: 2.0
    Exchange type: IKE_SA_INIT (34)
  ▷ Flags: 0x08 (Initiator, No higher version, Request)
    Message ID: 0x00000000
    Length: 1100
  ▷ Payload: Security Association (33)
  ▷ Payload: Key Exchange (34)
  ▷ Payload: Nonce (40)
  ▷ Payload: Notify (41) - NAT_DETECTION_SOURCE_IP
  ▷ Payload: Notify (41) - NAT_DETECTION_DESTINATION_IP
  ▷ Payload: Notify (41) - SIGNATURE_HASH_ALGORITHMS
```

IKEv2实例：消息1中的SA载荷

```
▲ Payload: Security Association (33)
    Next payload: Key Exchange (34)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 700
  ▲ Payload: Proposal (2) # 1
      Next payload: Proposal (2)
      0... .... = Critical Bit: Not Critical
      .000 0000 = Reserved: 0x00
      Payload length: 44
      Proposal number: 1
      Protocol ID: IKE (1)
      SPI Size: 0
      Proposal transforms: 4
    ▷ Payload: Transform (3)
    ▷ Payload: Transform (3)
    ▷ Payload: Transform (3)
    ▷ Payload: Transform (3)
  ▷ Payload: Proposal (2) # 2
  ▷ Payload: Proposal (2) # 3
  ▷ Payload: Proposal (2) # 4
```

SA载荷包含4个proposal

Proposal载荷包含4个T载荷

Proposal载荷下的T载荷

```
Proposal transforms: 4
▲ Payload: Transform (3)
    Next payload: Transform (3)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 12
    Transform Type: Encryption Algorithm (ENCR) (1)
    Reserved: 00
    Transform ID (ENCR): ENCR_AES_CBC (12)
  ▷ Transform Attribute (t=14,l=2): Key Length: 128
▲ Payload: Transform (3)
    Next payload: Transform (3)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 8
    Transform Type: Integrity Algorithm (INTEG) (3)
    Reserved: 00
    Transform ID (INTEG): AUTH_HMAC_SHA1_96 (2)
```

加密算法

完整性算法

```
▲ Payload: Transform (3)
    Next payload: Transform (3)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 8
    Transform Type: Pseudo-random Function (PRF) (2)
    Reserved: 00
    Transform ID (PRF): PRF_HMAC_SHA1 (2)
▲ Payload: Transform (3)
    Next payload: NONE / No Next Payload  (0)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 8
    Transform Type: Diffie-Hellman Group (D-H) (4)
    Reserved: 00
    Transform ID (D-H): 2048 bit MODP group (14)
```

伪随机函数

D-H群

IKEv2实例：消息2

- Internet Security Association and Key Management Protocol
  - Initiator SPI: 0dda36cf9a109c86
  - Responder SPI: 5fa187810183f065
  - Next payload: Security Association (33)
  - ▷ Version: 2.0
  - Exchange type: IKE SA INIT (34)
  - ▷ Flags: 0x20 (Responder, No higher version, Response)
  - Message ID: 0x00000000
  - Length: 448
  - ▷ Payload: Security Association (33)
  - ▷ Payload: Key Exchange (34)
  - ▷ Payload: Nonce (40)
  - ▷ Payload: Notify (41) - NAT_DETECTION_SOURCE_IP
  - ▷ Payload: Notify (41) - NAT_DETECTION_DESTINATION_IP
  - ▷ Payload: Notify (41) - SIGNATURE_HASH_ALGORITHMS

IKEv2实例：消息2中的SA

```
▲ Payload: Transform (3)
    Next payload: Transform (3)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 12
    Transform Type: Encryption Algorithm (ENCR) (1)
    Reserved: 00
    Transform ID (ENCR): ENCR_AES_CBC (12)
  ▲ Transform Attribute (t=14,l=2): Key Length: 128
      1... .... .... .... = Format: Type/Value (TV)
      Type: Key Length (14)
      Value: 0080
      Key Length: 128
▲ Payload: Transform (3)
    Next payload: Transform (3)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 8
    Transform Type: Integrity Algorithm (INTEG) (3)
    Reserved: 00
    Transform ID (INTEG): AUTH_HMAC_SHA1_96 (2)
```

IKEv2实例：消息2中的SA

```
▲ Payload: Transform (3)
    Next payload: Transform (3)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 8
    Transform Type: Pseudo-random Function (PRF) (2)
    Reserved: 00
    Transform ID (PRF): PRF_HMAC_SHA1 (2)
▲ Payload: Transform (3)
    Next payload: NONE / No Next Payload  (0)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 8
    Transform Type: Diffie-Hellman Group (D-H) (4)
    Reserved: 00
    Transform ID (D-H): 2048 bit MODP group (14)
```

IKE_SA_INIT结束后

通信双方均能生成SKEYSEED，然后IKE SA所需的所有密钥均从其导出。用于加密的密钥称为SK_e和用于完整性保护的密钥SK_a从SKEYSEED导出。每个方向均有独立的SK_e和SK_a。除了SK_e和SK_a外，另一个值SK_d也从SKEYSEED导出，用于生成将来导出Child SAs的密钥材料。后续消息除了消息头部外，整体被加密且完整性保护。

密钥导出

For IKE SA:

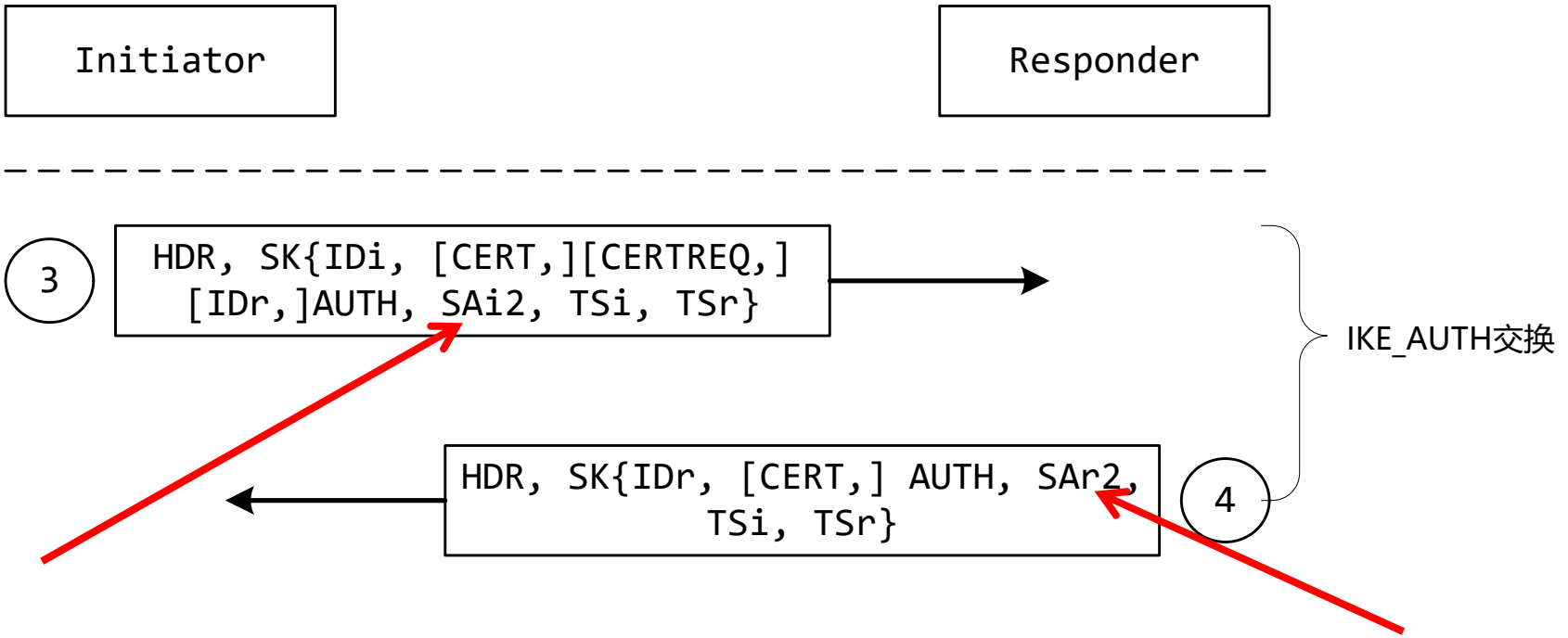
```
SKEYSEED = prf(Ni | Nr, g^ir)
```

```
{SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr}

 = prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr)
```

prf+( )函数

```
prf+ (K,S) = T1 | T2 | T3 | T4 | ...

   where:
   T1 = prf (K, S | 0x01)
   T2 = prf (K, T1 | S | 0x02)
   T3 = prf (K, T2 | S | 0x03)
   T4 = prf (K, T3 | S | 0x04)
   ...
```

IKE_AUTH交换

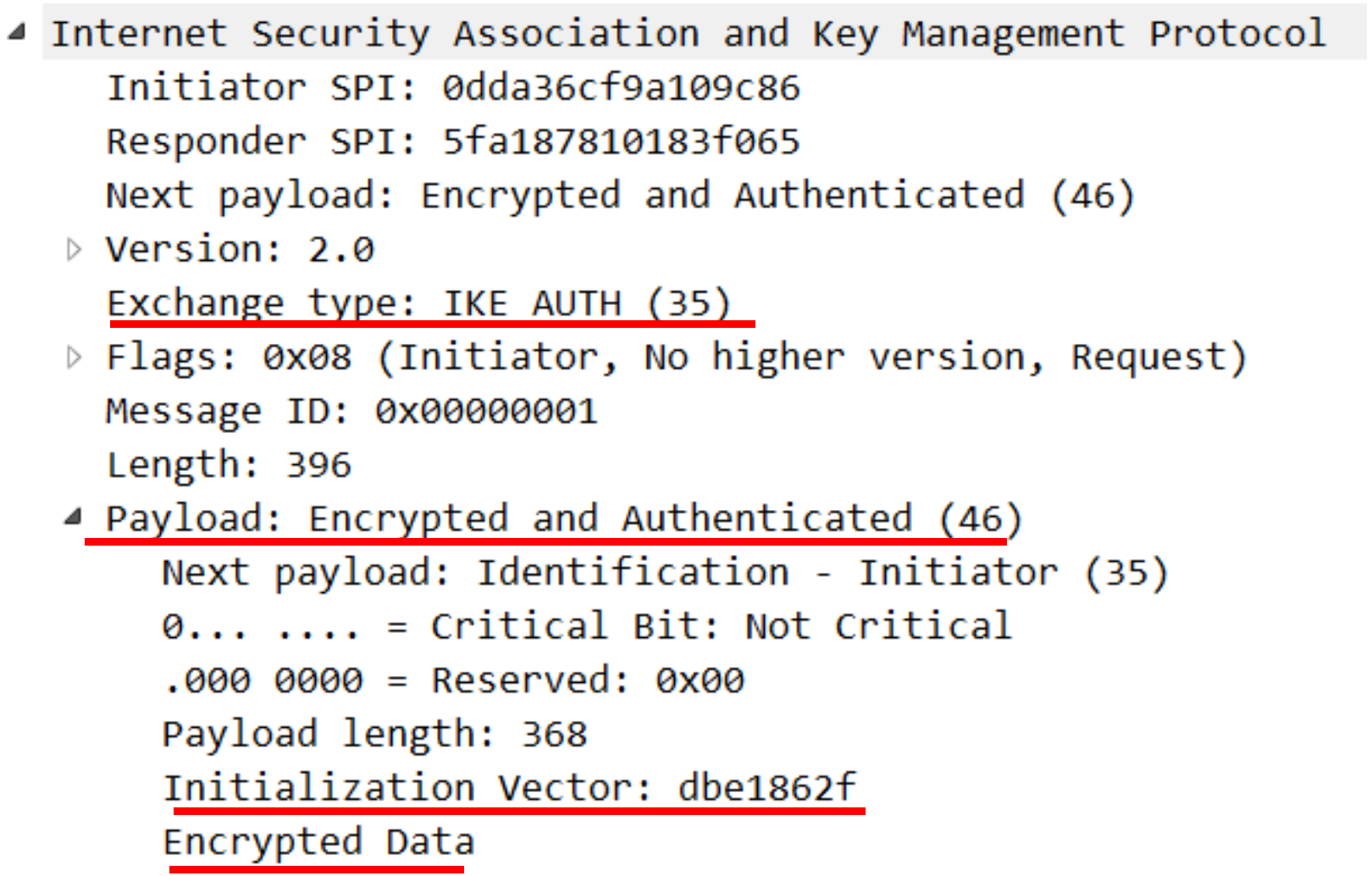


IKE_AUTH交换：负责身份认证，并创建第一个Child SA（一对IPSec SA）

IKE_AUTH

- 常用三种身份认证技术：
  - 预共享密钥方式：设备的身份信息为IP地址或名称
  - 数字证书方式：设备的身份信息为证书和通过证书私钥加密的部分消息Hash值（签名）
  - EAP方式
- 后续通过一对CREATE_CHILD_SA消息完成额外的IPSec SA创建

IKEv2实例：消息3

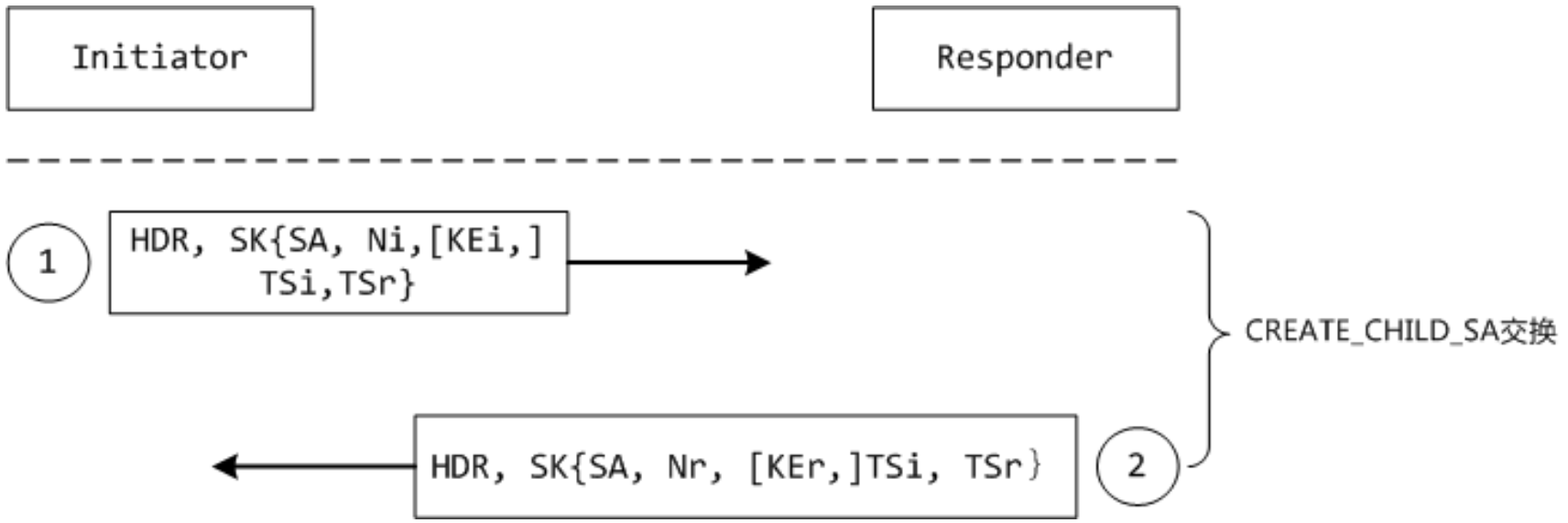

Internet Security Association and Key Management Protocol
Initiator SPI: 0dda36cf9a109c86
Responder SPI: 5fa187810183f065
Next payload: Encrypted and Authenticated (46)
Version: 2.0
Exchange type: IKE AUTH (35)
Flags: 0x08 (Initiator, No higher version, Request)
Message ID: 0x00000001
Length: 396
Payload: Encrypted and Authenticated (46)
Next payload: Identification - Initiator (35)
0... .... = Critical Bit: Not Critical
.000 0000 = Reserved: 0x00
Payload length: 368
Initialization Vector: dbe1862f
Encrypted Data

IKEv2实例：消息4

```
▲ Internet Security Association and Key Management Protocol
    Initiator SPI: 0dda36cf9a109c86
    Responder SPI: 5fa187810183f065
    Next payload: Encrypted and Authenticated (46)
  ▷ Version: 2.0
    Exchange type: IKE_AUTH (35)
  ▷ Flags: 0x20 (Responder, No higher version, Response)
    Message ID: 0x00000001
    Length: 220
  ▲ Payload: Encrypted and Authenticated (46)
      Next payload: Identification - Responder (36)
      0... .... = Critical Bit: Not Critical
      .000 0000 = Reserved: 0x00
      Payload length: 192
      Initialization Vector: a219fbc3
      Encrypted Data
```

CREATE_CHILD_SA交换

CREATE_CHILD_SA交换用于创建新的Child SA, rekey IKE SA



密钥导出

For CHILD SA:

```
KEYMAT = prf+(SK_d, Ni | Nr)
```

如果这个Child SA是第一个Child SA，则Ni和Nr是IKE_SA_INIT交换中Nonce

如果是通过后续交换创建的，则Ni和Nr是CREATE_CHILD_SA交换中Nonce信息。

密钥导出

For CHILD SA:

如果CREATE_CHILD_SA中包含可选的DH交换，则

KEYMAT的生成方式如下：

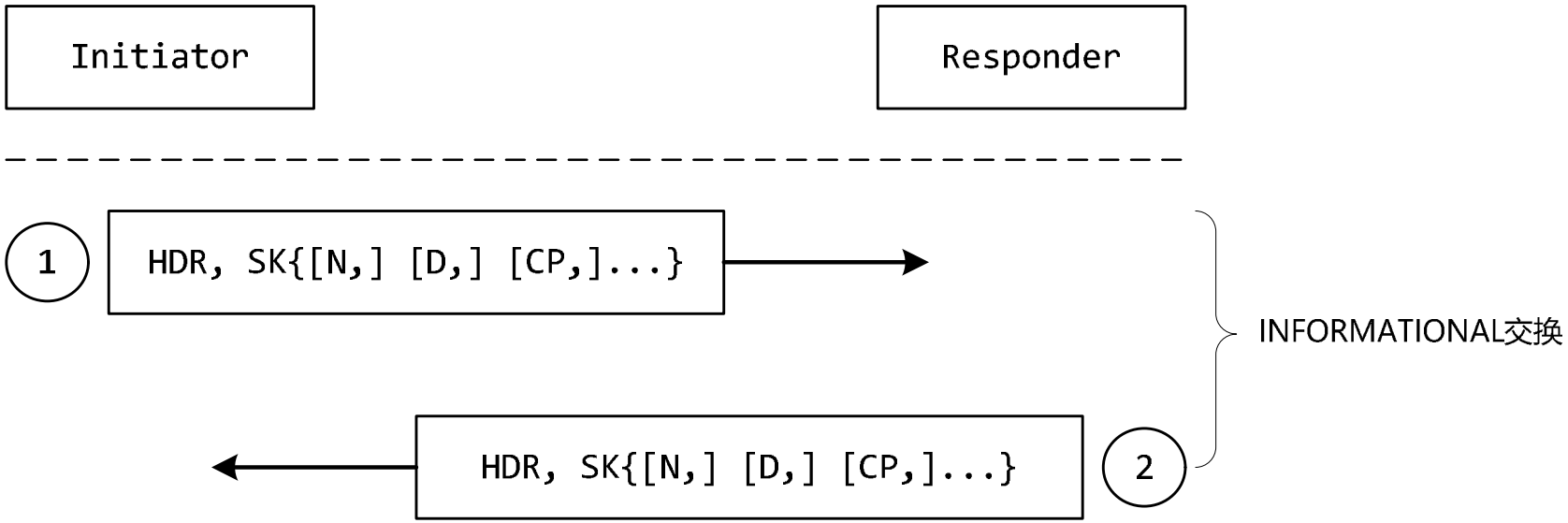
```
KEYMAT = prf+(SK_d, g^ir (new) | Ni | Nr)
```

Keymat分割规则

KEYMAT 包 含 了 IPSec SA 所 需 的 所 有 keys 。 KEYMAT是用prf+函数生成的，因此需要多长就生成多长。然后按照规则进行分割。

举个例子，对于ESP和AH，分割的顺序为：首先是加密密钥（其长度取决于协商的结果），然后是完整性密钥（如果有的话）。

INFORMATIONAL交换



INFORMATIONAL交换用于做housekeeping工作，比如删除SA、报告错误情况等

课后作业

- 分析wireguard的架构
  - https://www.wireguard.com/
- 分析noise protocol framework
  - https://noiseprotocol.org
  - Whatsapp、wireguard、lighting基于noise开发