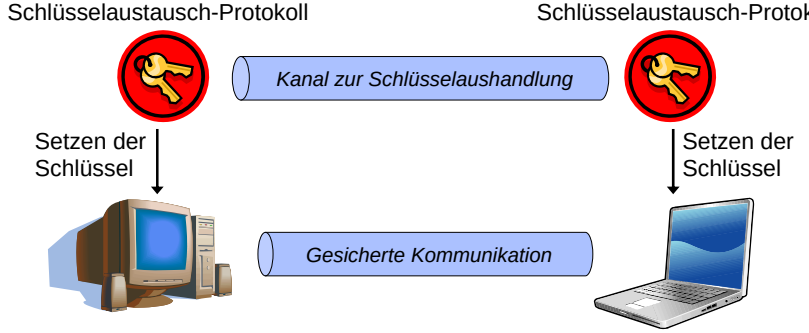
# Netzsicherheit – Architekturen und Protokolle
## Internet Key Exchange

1. Motivation
2. Bausteine des Schlüsselaustauschs
3. Internet Key Exchange

## Motivation




- Ziel: Gesicherte Kommunikation
- Schlüsselaustausch-Protokoll
  - Aushandlung der Austausch-/Sicherungsverfahren für den Kanal
  - Authentizitätsüberprüfung des Kommunikationspartners
  - Erzeugung gemeinsamer Schlüssel



•Aushandlung der Austausch-/Sicherungsverfahren für den Kanal
  •Welche Verfahren können beide Kommunikationspartner ausführen?
    •Schlüsselaustauschverfahren
    •Authentifizierungsverfahren
    •Ver-/Entschlüsselungsverfahren
    •Hash-Funktionen
  •Welche Verfahren werden bevorzugt?
•Authentizitätsüberprüfung des Kommunikationspartners
  •Wie kann die Authentizität des Kommunikationspartners überprüft werden?
    •Digitale Signaturen
    •Gemeinsames Geheimnis: z.B. Challenge Response Verfahren
•Erzeugung gemeinsamer Schlüssel für den Kanal
  •Wie wird das gemeinsame Geheimnis erzeugt?
    •Diffie-Hellman Austausch
    •Client- oder serverbasiert

## Einfacher statischer Ansatz

- Einigung über das verwendete Verfahren und Austausch des Schlüsselmaterials durch persönliche Übergabe
  - Verschlüsselung/Authentifizierung der auszutauschenden Daten mit dem erhaltenen Schlüssel

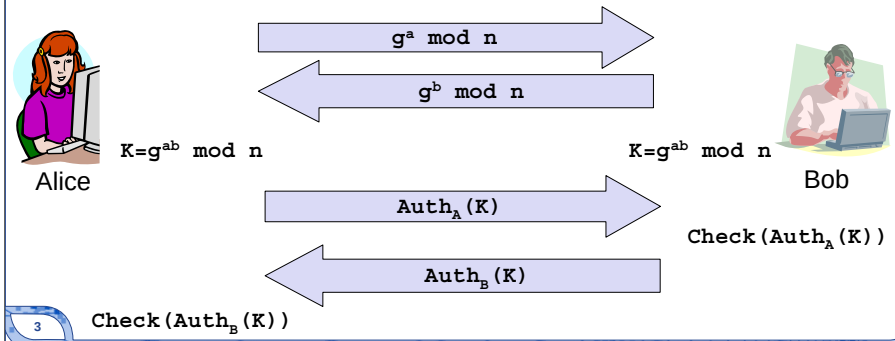
☺ sehr einfaches Verfahren
☺ Schlüssel ist automatisch authentifiziert

☹ „Persönliches Treffen“ notwendig
☹ Erneuerung der Schlüssel erfordert neues Treffen
☹ schlechte Skalierbarkeit
☹ Problem von langlebigen Schlüsseln
☹ keine dynamische Wahl des Verfahrens
☹ spätere Schlüsselpreisgabe legt auch die Kommunikation offen



## Erster einfacher dynamischer Ansatz

- Diffie-Hellman-Austausch mit Authentifizierung
  - Authentifizierung hier als **`Auth(…)`**
  - Voraussetzung: Diffie-Hellman-Gruppe und somit Generator und Modulus bekannt






•Für Grundlagen siehe Krypto-Grundlagen-Kapitel

•„Auth“ ist hier eine Authentifizierungsfunktion, die in der Lage ist das Argument (z.B. K) zu Authentifizieren, also Veränderungen durch Dritte bemerkbar zu machen. Achtung: AuthA heißt nur das Alice diesen Wert erzeugt, es ist nicht gesagt, dass AuthA != AuthB !

## Probleme mit dem Ansatz

- Probleme des Austausch-Protokolls
  - Verwendung von Konstanten für
    - *Diffie-Hellman-Gruppe* und somit
    - *Generator und Modulus*
  - Sitzungsschlüssel als Eingabe der Authentifizierung
    - schwache Authentifizierungsfunktion kann Information offenlegen
  - beide Authentifizierungsnachrichten sehr ähnlich
    - Erzeugung der Authentifizierungsnachricht ohne Kenntnis des Schlüssels möglich

→ es geht schneller, *3 Nachrichten* wären ausreichend

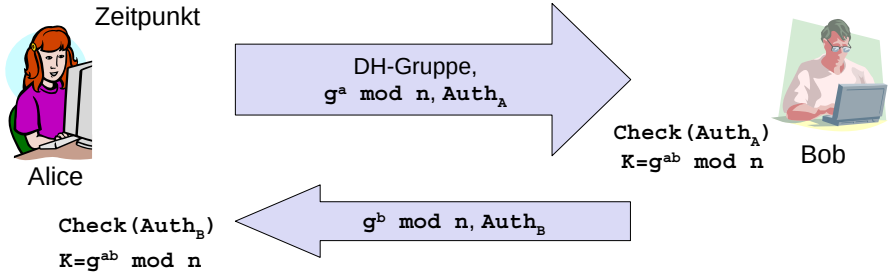
**Wie sieht das Protokoll mit drei Nachrichten aus?**



•Falls die beiden Auth-Nachrichten exakt die gleiche Bitfolge authentifizieren, müssen zwangsläufig die Schlüssel unterschiedlich sein, sonst wäre das Ergebnis der Authentifizierung auf beiden Seiten (Alice und Bob) gleich. Der Angreifer könnte dann also den zweiten Auth-Wert aus dem ersten ableiten (Identität).

## Zweiter dynamischer Ansatz

- Dynamische Wahl der Diffie-Hellman-Gruppe und des Generators `(g,n)`
  - Überprüfung der Gruppe durch Bob!
- Authentifizierung unabhängig vom Sitzungsschlüssel
  - **`Auth`**-Funktion über alle Nachrichten und Felder bis zu diesem Zeitpunkt






•In AuthA fließen nur Eingaben von Alice ein und nicht von Bob. Somit wird nur ein Teil der Eingaben für K von Alice authentifiziert.

## Probleme mit dem zweiten Ansatz

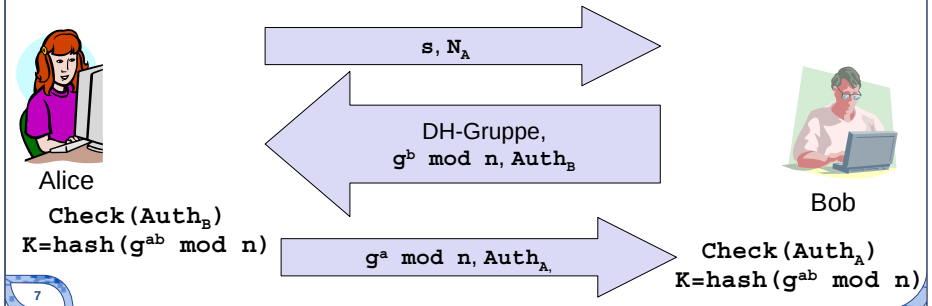
- Probleme des Austausch-Protokolls
  - Gewählte DH-Gruppe für Bob zu schwach
    - Abbruch des Protokolls
    - Optional: Fehlermeldung an Alice
    - Neustart des Protokolls mit neuer DH-Gruppe
  - Wiederholungsangriffe
    - Angreifer wiederholt abgefangenes Paket von Alice
    - Bob führt das Protokoll weiter aus (DoS)
    - Angreifer kann den Sitzungsschlüssel nicht lernen
    - Aber: Eintrag einer erfolgreichen Authentifizierung im Log von Bob
      - fehlerhafte Informationen bei späterer Fehlersuche
  - Variabel langer Schlüssel als Ergebnis des DH-Austauschs
    - Blockchiffren benötigen aber feste Schlüssellänge



•Ein weiterer Nachteil: Identitäten von A und B können einfach passiv mitgehört werden…

## Dritter dynamischer Ansatz

- Minimale Sicherheitsanforderung
  - Alice gibt eine Unterschranke $s$ für Diffie-Hellman-Gruppe an
- Schutz vor Wiederholungsangriffen
  - Nonce $N_X$ (Number only used once)
  - **Auth**-Funktion über alle Nachrichten und Felder bis zu diesem Zeitpunkt (und somit auch die Nonce)






•Auch hier: Identitäten können passiv mitgehört werden

•Frage: Wofür ist die Nonce?

- •Antwort: Schutz gegen Wiederholungsangriffe.
- •Kann zusätzlich noch in die Berechnung des Schlüsselmaterials eingehen.
- •Wiederverwendung von gb bzw gb mod n dann weniger unsicher.

# Gefahren für Austauschprotokolle

- Wiederholungsangriffe (Replay Attack)
  - Wiederholung von zuvor aufgezeichneten Nachrichten
- Denial-of-Service-Angriffe (Denial of Service Attack)
  - Erschöpfen einer physischen (Speicher oder CPU-Zeit) oder einer virtuellen Ressource (Zustände)

- Downgrade Attack
  - Löschen von starken Algorithmen aus der Liste der unterstützten Verfahren
  - Angreifer schafft es, dass ein schwaches Verfahren gewählt wird
- Verkürzungsangriff (Truncation Attack)
  - ungesichertes (einseitiges) Beenden einer Verbindung durch Dritten
  - ein Teilnehmer könnte mehr Daten gesendet haben als der andere empfangen hat
- Schlüsselhinterlegung bei einer vertrauenswürdigen Organisation
  - Missbrauch des hinterlegten Schlüssels

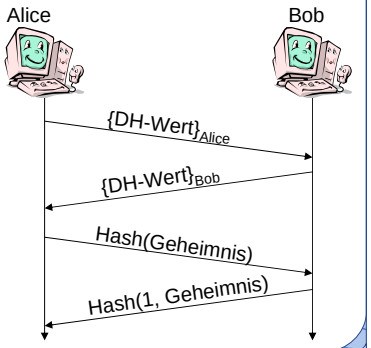


# Netzsicherheit – Architekturen und Protokolle
## Internet Key Exchange

# Perfect Forward Secrecy

- Perfect Forward Secrecy (PFS)
  - Langlebige Schlüssel sind unabhängig vom Sitzungsschlüssel
  - Vernichtung des Sitzungsschlüssel nach Beendigung der Kommunikation
  - Aufzeichnen aller Nachrichten und Einbruch in die Endsysteme führt nicht zur Offenlegung der Kommunikation
  - wird auch in dynamischen Schlüsselaustauschprotokollen verwendet
- Schlüsselhinterlegung bei vertrauenswürdiger Organisation
  - Schlüsselwiederherstellung
  - Problem: Missbrauch des hinterlegten Schlüssels

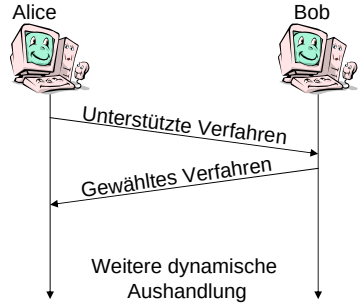
  → PFS-Algorithmen können vor dieser Schwachstelle schützen






# Dynamische Wahl der Verfahren

- Verhandlung der Sicherungsmechanismen
  - einfache Migration zu kryptographisch stärkeren Verfahren
  - Ausschluss gebrochener Verfahren
  - keine Festlegung durch das Standardisierungsgremium notwendig
    - Verfahren für Interoperabilität




- Problem 1: Komplexität des Protokolls
  - wie werden Verfahren beschrieben?
  - welche Kombinationen sind zulässig?
- Problem 2: Angreifer löscht die Verfahren, die er nicht brechen kann
  - Verkürzungsangriff: Downgrade Attack
  - Alice und Bob haben noch kein gemeinsames Geheimnis, um die Nachrichten zu schützen

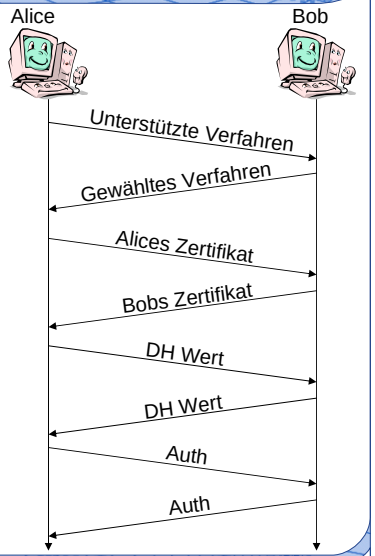


•Problem 2

- •Ein Verfahren könnte z.B. NULL-Verschlüsselung (keine Verschlüsselung) sein.
- •Sollte nur noch dieses Verfahren übrig bleiben, kann der Angreifer einfach mitlesen.

## Verhinderung von Downgrade-Angriffen

- Ziel: Erkennung von Änderungen an Nachrichten
  - Löschen kryptographisch starker Algorithmen
- **`Auth`**-Funktion über alle gesendeten Nachrichten und Felder
  - bis jetzt noch keine vertraulichen Daten gesendet
  - bei Nichtübereinstimmen von empfangenem und berechnetem Authentifizierungswert wird Schlüsselmaterial ungültig gemacht, Verfahren abgebrochen



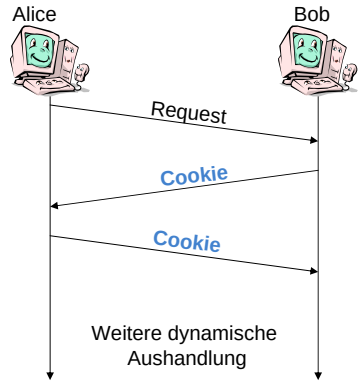



•Mehrere Reihenfolgen sind denkbar. Bsp.: Zertifikat- und DH-Austausch können vertauscht sein.

•8 Nachrichten sind natürlich etwas ineffizient, 4 wären einfach machbar.

## Cookies und Puzzles

- Problem: Angreifer kann Nutzer durch Erschöpfen einer physischen (Speicher) oder einer virtuellen Ressource (Zustände) aussperren
- Cookies
  - Berechnung von Cookies aus lokalen Daten und Paketdaten
  - Ziel
    - keinen lokalen Zustand halten
    - Erkennung gefälschter Absenderadressen
- Puzzles
  - Stellen einer rechenintensiven Aufgabe an Stelle des Cookies
  - Abhängig von Anzahl Requests
  - wieder keine Zustandserzeugung



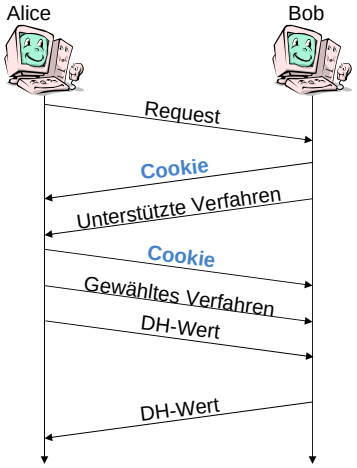



•Der Cookie enthält Zustand, der dann nicht lokal (auf Bob) gehalten werden wird. Somit ist ein Denial-of-Service-Angriff auf Bobs Speicher schwieriger.

•Falls Alice das Cookie zurückschickt, ist mit hoher Wahrscheinlichkeit Alices Adresse nicht gefälscht.

# Einschränkung von DoS-Angriffen

- Ziel: keinen lokalen Zustand vor der Überprüfung Alices Adresse!
- Zustandslose Cookies
  - Codierung des Zustands in Cookie
  - Z.B. Hash über langlebiges Geheimnisse + Nachrichtendaten
- Rechenintensive Operationen werden zuerst vom Anfragenden ausgeführt
  - Zertifikatsüberprüfung
  - Diffie-Hellman-Berechnung
  - Signieren des Diffie-Hellman-Werts



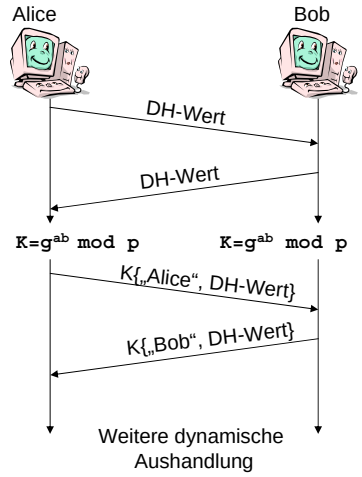



•Vermeidung von Denial-of-Service-Angriffen gegen die Rechenzeit (CPU) von Bob, da DH, RSA, usw. auf kleinen Geräten teuer sind.

•Man kann in die zustandslosen Cookies entweder 1 Geheimnis eingehen lassen (als Schutz vor Fälschung des Cookies) oder ein Geheimnis aus einer Liste auswählen, dann ist aber ein Index notwendig. Die vereinfacht es die Geheimnisse auszutauschen („rekeying“). Neues Geheimnis → neuer Index

# Schutz der Identitäten

- Problem: passiver Angreifer kann die Identitäten der Kommunikations-partner abhören
- Lösung
  - Anonymer Diffie-Hellman Austausch
  - Übertragung der Identität geschützt durch den ausgetauschten Schlüssel
  - Übertragung des signierten DH-Werts
- Problem: kein Schutz vor aktiven Angreifern (Man-in-the-Middle)
- Lösung: nur Kenntnis des öffentlichen Schlüssels vor dem Austausch schützt beide Identitäten



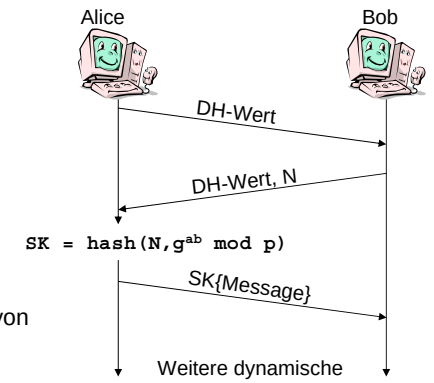



•K{N}: N wird verschlüsselt übertragen. Als Schlüssel wird K genutzt.

•Im Beispiel wird „DH-Wert“ als Schlüssel verwendet!

# Live Partner Reassurance

- Wiederverwendung von Diffie-Hellman-Werten
  - → rechenintensive Arbeit so selten wie möglich durchführen
- Problem: Wiederholungsangriffe
- Lösung: Nonces
  - beliebige Zufallszahl (Nonce)
  - für jede Verbindung eine andere
  - gemeinsames Geheimnis hängt ab von
    - Diffie-Hellman-Austausch
    - Nonce
    - → `SK`: Seeded Key
  - Erkennung von Wiederholungsangriffen auch bei Wiederverwendung des Diffie-Hellman-Werts






•SK{N}: N wird verschlüsselt übertragen.

•→ Seeded Key: Also Schlüssel K und Seed=Nonce N parametrisieren das Verfahren

# Schlüsselerneuerung

- Aushandlung neuer Schlüssel notwendig, wenn
  - Lebenszeit des Schlüsselmaterials abgelaufen ist
  - maximal zu schützende Datenmenge gesichert wurde
  - Anti-Replay-Counter überläuft
    - Sequenznummer zur Erkennung von Duplikaten
  - neue Attribute für das Schlüsselmaterial benötigt wu

- Neuaushandlung der Schlüssel
  - Schlüsselerneuerung für Schlüsselaustausch-Kanal
  - Schlüsselerneuerung für Datenaustausch-Kanal
  - DH-Austausch, wenn PFS gefordert ist
    - Unabhängigkeit von langlebigem und aktuellem Schlüssel

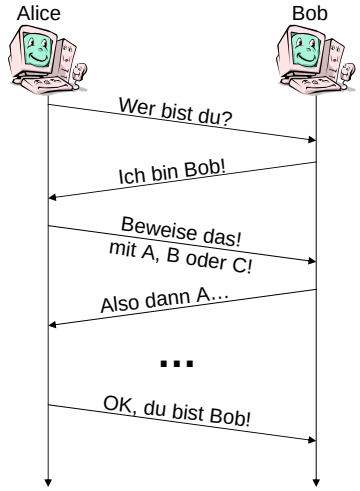


•Oft gilt: „die maximal zu schützende Datenmenge gesichert wurde“ = „der Anti-Replay-Counter überläuft“

•Sequenznummer werden im Kapitel IPsec betrachtet

## Modulare Authentifizierung

- Authentifizierung
  - Kommunikationspartner *beweist* Identität
  - Beispiele für Verfahren
    - Passwort, PIN, Schlüssel
    - Zertifikat, Smart-Card, SIM-Karte
- Modulare Authentifizierung
  - Entkopplung des Schlüsselaustauschverfahren vom Authentifizierungsverfahren
  - zukünftige Verfahren können einfach integriert werden: Kompatibilität
  - Authentifizierung kann auch durch Dritten erfolgen, siehe Kapitel Infrastrukturschutz
- EAP ist ein Protokoll zur modularen Authentifizierung





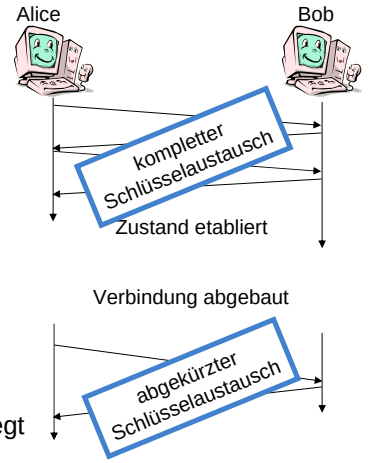

•Motivation

- •Standardisierung und Implementierung von Schlüsselaustausch-/Sicherheitsprotokollen von Authentifizierung entkoppeln
- •Dynamisch adaptierbar
  - •verschiedene Einsatzszenarien (mobile vs. stationär)
  - •verschiedene Nutzer und Systeme (Handy vs. PC)

•Ein Protokoll für modulare Authentifizierung ist das „Extensible Authentication Protocol“ (EAP), welches im Kapitel „Infrastrukturschutz“ vorgestellt wird. Es findet Anwendung in z.B. IKE (siehe nächsten Abschnitt).

## Sitzungswiederaufnahme

- Sitzungswiederaufnahme
  - *session resumption*
  - Schlüsselaushandlung teuer
    - nach kurzer Pause kürzt man den Aufwand für DH, EAP, RSA, …
  - Ansatz 1: zustandsbehaftet
    - Bob speichert Zustand, Alice bekommt i.d.R. nur ID
  - Ansatz 2: zustandslos (für Bob)
    - Bob kodiert kompletten Zustand in geschütztes Cookie o.ä.
    - Alice speichert den Zustand und legt ihn wieder vor
    - bessere DoS-Resistenz






•Die Motivation für dieses Verfahren ist es, dass ein Schlüsselaustausch 2 oder mehr Round-Trip-Times (RTTs) benötigt. Bei modularer Authentifizierung sogar noch einige mehr. Daher möchte man dies möglichst selten durchführen. Sollte es jetzt zu einem Abbruch der ursprünglichen Verbindung kommen (Netz beim Client kurz weg), dann ist es vorteilhaft diese Aushandlungen sich zu sparen.

•Beispiele hierfür

- •TLS (zustandsbehaftet) [siehe TLS-Kapitel]
- •RFC 4507 „Transport Layer Security (TLS) Session Resumption without Server-Side State“, 2006.
- •Zustandsloses Verfahren für IPsec derzeit in der Diskussion bei der Internet Engineering Task Force (IETF)

# Welche vorgestellten Bausteine implementiert Kerberos (nicht)?



- Kerberos bietet keine Perfect Forward Secrecy, keine dynamische Aushandlung der Verfahren, keinen Schutz vor DoS-Angriffen und keinen Schutz der Identitäten. Auch modulare Authentifizierung wird nicht unterstützt.
- Live Partner Reassurance durch Zeitstempel im Authenticator und Schlüsselerneuerung durch Anforderung eines neuen Tickets möglich
- Schutz des Sitzungs-Schlüssels durch Master-Secret des Benutzer
  - Erzeugung des Master-Secrets aus Passwort
  - Neues Master-Secret erst nach Änderung des Passworts

## Netzsicherheit – Architekturen und Protokolle
## Internet Key Exchange

## Internet Key Exchange

- Sichere Aushandlung von IPsec-Parametern
  - IKEv1 wurde in drei „Standard Track“ RFCs spezifiziert
  - IKEv2 spezifiziert in RFC 4306 (Dez. 2005)
- Ziele von Internet Key Exchange
  - *Aufbau eines gesicherten Kanals*
    - ISAKMP (Internet Security Association and Key Management Protocol)
      - Baukasten für Parameter-Aushandlungs-Protokolle
      - Definiert Format für Dateneinheiten
    - gegenseitige Authentifizierung, Diffie-Hellman-Austausch
    - Aufbau einer *IKE-SA* (oder auch ISAKMP-SA)
  - *Aushandlung des IPsec-Schlüsselmaterials*
    - Wahl der zu verwendenden Verfahren
    - Generierung der Schlüssel



## IKEv2 Überblick

- IKEv2 wurde als Ersatz für IKEv1 entworfen
  - IKEv1 sehr komplex und umstritten
  - IKEv1 nur Authentifizierung mit Public-Key oder Pre-Shared-Secret
    - Erweiterungen wie das unsichere XAUTH (siehe Cisco VPN) waren die Folge, um Authentifizierung mit Passwort zu unterstützen
- Verbesserungen in IKEv2
  - geringere Komplexität
    - nur noch ein Modus, IKEv1 hatte 8 Modi
  - Unterstützung von Cookies für DoS-Resistenz
  - Geringere Latenz beim Aufbau (2 Umlaufzeiten)
  - Konfigurations-Daten können getunnelt werden
  - besserer Umgang mit NAT-Gateways
  - modulare Authentifizierung per EAP



•Round Trip Times (RTTs): Umlaufzeiten, die Zeit die ein Paket von Alice zu Bob und zurück braucht.

## Einschub: Definitionen

- Security Association (SA)
  - Sicherheitsassoziation zwischen zwei Kommunikationspartnern
  - repräsentiert Verbindung und Zustand
  - kryptographisches Material (Schlüssel, IV, …)
  - anhand des SPI selektierbar
    - SPI: Security Parameter Index
    - Index in einer Tabelle von Einträgen

- Traffic Selector (TS)
  - Neuerung in IKEv2
  - beschreibt den zu schützenden Verkehr
  - notwendig, um Verkehr einer SA zuzuordnen
    - z.B: `TCP / IP1:Port1` → `IP2:Port2`
    - z.B: `UDP / IP-Range:*` → `IP-Range:*`



•Traffic Selector beschreibt den Traffic, welcher geschützt werden soll.

•z.B. TCP / 192.168.0.1 – 192.168.0.10 Port 80-82 ←→ 10.13.0.1 – 10.13.0.100 Port 1000-2000

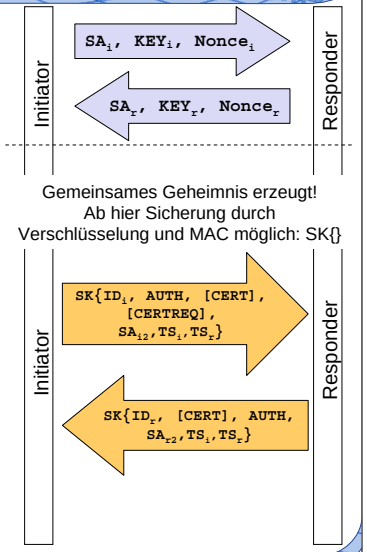
## Initialer Austausch

- `IKE_SA_INIT` – Nachricht 1 und 2
  - Wahl/Auswahl von Algorithmen (SA)
  - Diffie-Hellman-Austausch (KEY)
  - Anschließend kann die „IKE-SA“ aufgebaut werden
    - Sicherung und Verschlüsselung möglich

- `IKE_AUTH` – Nachricht 3 und 4
  - Überprüfen der Identitäten (ID)
  - Authentizieren des DH-Austausches
  - Aushandlung der „Child-SA“ ($SA_{i2}$/$SA_{r2}$)
  - Austausch von Traffic-Selectoren ($TS_i$, $TS_r$)
  - Austausch von Zertifikaten (CERTREQ, CERT)

- ISAKMP-Payloads enthalten

| | |
|---|---|
| SA: | Auswahl bzw. gewähltes Verfahren |
| KEY: | Diffie-Hellmann Wert (DH) |
| Nonce: | Zufallswert |
| ID: | Identitäten der Kommunikationspartner |
| CERT: | Verwendetes Zertifikat |
| CERTREQ: | Anforderung von Client-Zertifikat |
| AUTH: | Signierter Hash der Austausch-Nachrichten |
| TS: | Traffic Selector |
| […] | bedeutet optionaler Payload … |






•siehe auch RFC4306 Kapitel 1.2

•Bemerkung: Wir verwenden SK{} statt nur K{} bei IKE, da diese Funktion nicht nur mit einem Schlüssel, sondern auch mit einem Seed parametrisiert wird.

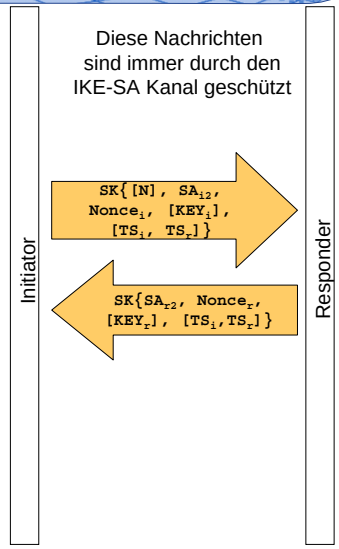
•Zusatzinformationen

- •Der Seed wird aus den Nonces berechnet
- •SEED = prf ( Ni | Nr , gir)  (prf: pseudo-random function)

# Aushandlung von Child-SAs

- Falls IKE-SA schon aufgebaut ist, kann man eine Child-SA auch direkt aushandeln
  - für Rekeying oder weitere SAs
  - früher als Phase 2 (Quick Mode) bezeichnet (IKEv1)
- CREATE_CHILD_SA (Nachricht 1 & 2)
  - eventuell zusätzlicher DH-Austausch
  - Aushandlung der „Child-SA“ ($SA_{i2}/SA_{r2}$)
    - hier wieder Vorschläge und Auswahl
  - Traffic-Selectoren, falls neue SA
    - bei Rekeying nicht notwendig
- ISAKMP-Payloads enthalten

| | |
|---|---|
| SA: | Auswahl bzw. gewähltes Verfahren |
| KEY: | Diffie-Hellmann Wert |
| Nonce: | Zufallswert, gehen in Schlüsselgen. ein |
| N: | Notify (Benachrichtigungs-Payload) |
| TS: | Traffic Selector |





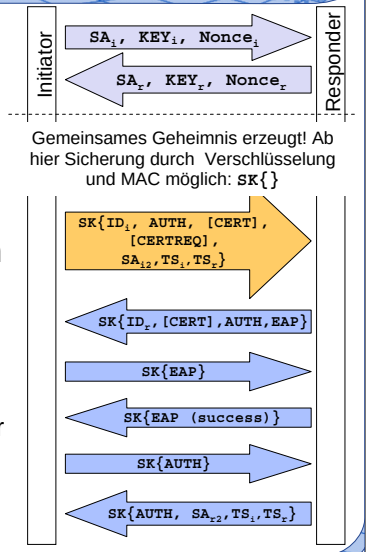

•Notify-Payload kann Fehler und Statusmeldungen anzeigen. (z.B. IPCOMP_SUPPORTED, NAT_DETECTION_SOURCE_IP, REKEY_SA, …)

•IPCOMP: IP Compression – Kompression von IP Daten

•NAT: Network Address Translation – Umsetzen von IP-Adressen. Meistens werden private (RFC 1918) Adressen auf eine oder mehrere öffentliche routebare Adressen umgesetzt. z.B. 192.168.0.x → 141.3.70.1

# Initialer Austausch und EAP

- Nachrichten 1-3 wie bisher
  - ab Nachricht 4 unterschiedlich
- Nachrichten 4-7 modulare Authentifizierung mittels EAP
- Nachricht 8 schließt Aushandlung von Client-SA ab
- Motivation
  - Extensible Authentication Protocol (EAP) ermöglicht nahezu beliebige Protokolle zur Authentisierung
  - Authentication „Plug-and-Play“
  - siehe Kapitel Infrastrukturschutz





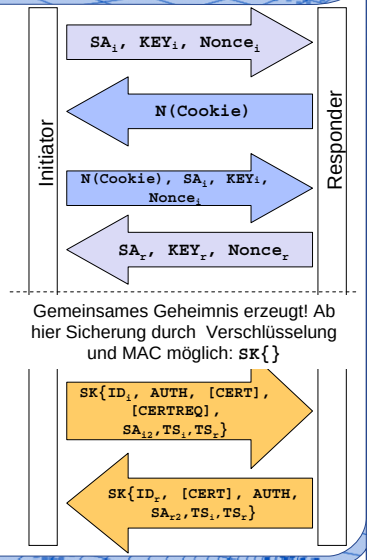

•Extensible Authentication Protocol (EAP):

- •sehr verbreitetes Verfahren für modulare Authentifizierung
- •unterstützt z.B. passwortbasierte, zertifikatsbasierte und SIM-basierte Authentifizierung
- •definiert in RFC3748
- •siehe auch Kapitel Infrastrukturschutz in dieser Vorlesung

•Wichig: AUTH ist hier natürlich nicht unbedingt ein signierter Hash, da die Authentifizierung (zumeist des Initiators gegenüber dem Responder) per EAP gemacht wird. Die Responder -> Initiator Authentifizierung kann allerdings wie bisher ein signierter Hash sein.

# Initialer Austausch und DoS

- Nachricht 1 wie bisher
- Falls Responder einen DoS-Angriff vermutet
  - Nachricht 2: Notify (N) mit Cookie
  - Nachricht 3: Initiator muss mit Cookie antworten
    - Cookie: 1-64 Oktette lang, von Responder beliebig wählbar (siehe nächste Folie)
- sonst: wie bisher
- Motivation:
  - Zustand muss erst nach Nachricht 3 gehalten werden
  - DoS wird wesentlich schwieriger!
    - denn Spoofing ist nicht mehr möglich und eine Nachricht reicht nicht mehr






•siehe auch RFC4306 Kapitel 1.2

•Traffic Selector beschreibt den Traffic, welcher geschützt werden soll.

•z.B. TCP / 192.168.0.1 – 192.168.0.10 Port 80-82 ←→ 10.13.0.1 – 10.13.0.100 Port 1000-2000

# Initialer Austausch und DoS (2)

- Erzeugen vom Cookie, Vorschlag nach IKEv2
  - `Cookie = <ID> | Hash(N_i | IP_i | SPI_i | <secret>)`
- Tabelle von Geheimnissen

| ID | Secret |
|---|---|
| 1057 | 4711 |
| 1058 | 1234 |
| 1059 | 3344 |

- Vorteile
  - Ändern vom Geheimnis während Angriffen möglich
  - Angreifer muss derzeit gültige ID raten
    - falls ID 32bit lang: x aus $2^{32}$, im Beispiel x=3



•Internet Key Exchange v2 (IKEv2): RFC4306, http://www.rfc-editor.org/rfc/rfc4306.txt , 2.6

•ID: Index des Geheimnisses

•Ni: Nonce des Initiators

•IPi: IP des Initiators

•SPIi: SPI des Initiators

•Beispiel: Cookie = 1057 | Hash (456 | 192.168.0.1 | 123 | 4711)

## Konfigurationsdaten über IKEv2

- IKEv1 konnte keine Konfigurationsdaten transportieren
  - man sollte DHCP o.ä. verwenden
  - aber was passiert, wenn für eine Sicherheitsassoziation (SA) kein DHCP verwendet werden kann/soll?
- Neuerung bei IKEv2
  - Configuration Payload (CP)
  - Pull-Verfahren
    - Teilnehmer sendet **`CFG_REQUEST`**
    - Gegenseite antwortet mit **`CFG_REPLY`**
  - Push-Verfahren
    - Teilnehmer sendet **`CFG_SET`** um Einstellung zu machen
    - Antwort per **`CFG_ACK`** oder Notify für Fehler



•Motivation

- •mehr Kontrolle für den Administrator
- •stabileres Verfahren als DHCP über IPsec
- •nutzerabhängige Konfiguration möglich (z.B. für gleichen Nutzer immer gleiche Adresse → Firewalling)

## Beispiel Attribute

| Attribute Type | Value | Multi-Valued | Length |
|---|---|---|---|
| RESERVED | 0 | | |
| INTERNAL_IP4_ADDRESS | 1 | YES* | 0 or 4 octets |
| INTERNAL_IP4_NETMASK | 2 | NO | 0 or 4 octets |
| INTERNAL_IP4_DNS | 3 | YES | 0 or 4 octets |
| INTERNAL_IP4_NBNS | 4 | YES | 0 or 4 octets |
| INTERNAL_ADDRESS_EXPIRY | 5 | NO | 0 or 4 octets |
| INTERNAL_IP4_DHCP | 6 | YES | 0 or 4 octets |
| APPLICATION_VERSION | 7 | NO | 0 or more |
| INTERNAL_IP6_ADDRESS | 8 | YES* | 0 or 17 octets |
| RESERVED | 9 | | |
| INTERNAL_IP6_DNS | 10 | YES | 0 or 16 octets |
| INTERNAL_IP6_NBNS | 11 | YES | 0 or 16 octets |
| INTERNAL_IP6_DHCP | 12 | YES | 0 or 16 octets |
| INTERNAL_IP4_SUBNET | 13 | YES | 0 or 8 octets |
| SUPPORTED_ATTRIBUTES | 14 | NO | Multiple of 2 |
| INTERNAL_IP6_SUBNET | 15 | YES | 17 octets |

* These attributes may be multi-valued on return only if multiple values were requested.



•Zusatzinformation! (Für die Prüfung bitte nicht alle Attribute auswendig lernen.)

•NBNS → NetBios Name Server (WINS)

•Quelle und weitere Erklärungen: RFC 4306, Seite 82-84

- •http://www.ietf.org/rfc/rfc4306.txt oder http://www.rfc-editor.org/cgi-bin/rfcdoctype.pl?loc=RFC&letsgo=4306&type=http&file_format=pdf

# NAT-Traversal

- Network Address Translation (NAT) ändert IP-Adressen/Ports

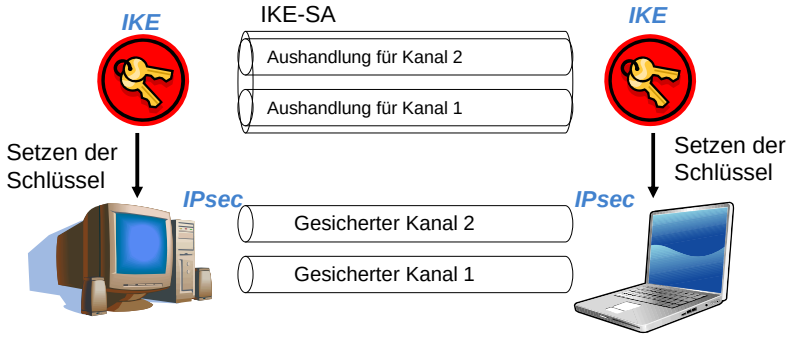

- Lösung: IKEv2 transportiert mittels Notify-Payload den Hash der Adressen (SPI, IP, Port) für jeweils beide Seiten (Initiator und Responder)
- daraufhin kann Gegenseite NATs detektieren



•SPI: Security Parameter Index – der Index in der Tabelle der Sicherheit Assoziationen (SAs)

•Es reicht aus, den Hash zu übertragen, da man nur testen will, ob sich etwas geändert hat. Sollte dies der Fall sein, kann man immer noch mehr übertragen. Die Motivation waren sicherlich IPv6-Adressen, welche relativ lang sind und somit viel Overhead bedeutet hätten.

# IKE Zusammenfassung




- Zusammenfassung der Phasen
  - IKE-SA schützt Aushandlung weiterer Schlüssel
  - Child-SA stellt Schlüsselmaterial für Anwendung dar (IPsec)
  - Anwendung (IPsec) sichert den Datenaustausch

# Überprüfen Sie, welche vorgestellten Bausteine in IKEv2 verwendet werden!

Wie wurden die Mechanismen integriert?





- Fahrplan
  - 01.07.09 Infrastrukturschutz-1
  - 08.07.09 Infrastrukturschutz-2
  - 15.07.09 Privilege Management Infrastructure

- 22.07.09 → Wiederholungs-Vorlesung
  - welche Themen sollen noch einmal wiederholt werden?
  - wo habt ihr noch Fragen? Unklarheiten?
  - bitte per Email an mayer@tm.uka.de
  - Vorschläge/Fragen bis spätestens diesen Freitag!

# Literatur

Bücher (beziehen sich noch auf RFC240x-IPsec von 1998)

- S. Frankel; Demystifying the IPsec Puzzle; Artech House, 2001
  - gutes IPsec-Buch (noch IKEv1)
- C. Kaufmann, R. Perlman, M. Speciner; Network Security – Private Communication in a public world; Prentice Hall; 2003
  - allgemeineres Buch

Standards und Papers

- RFC4301 – RFC4308 Dez. 2005 IPsec Standards, IETF
  - aktuelle Standards
- Ferguson, N und Scheier, B.; A Cryptographic Evaluation of IPsec“, http://www.counterpane.com/ipsec.html, Feb. 1999
- Simpson, W.; IKE/ISAKMP Considered Dangerous; Draft; Jun. 1999
- RFC 2401 – RFC 2409 1998 IPsec Standards, IETF
  - veraltete Standards