ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Dipartimento di Informatica - Scienza e Ingegneria

Corso di Laurea in Ingegneria e Scienze Informatiche

Sicurezza del protocollo BGP: attacchi e contromisure nella letteratura scientifica

Relatore:
Chiar.mo Prof.
Andrea Piroddi

Presentata da:
Roberto Pisu

Sessione Ottobre 2025
Anno Accademico 2024/2025

(DA FARE ALLA FINE)
5 parole chiave per caratterizzare il contenuto della dissertazione:
(se non ti piacciono così sparse puoi anche semplicamente scriverle su una riga sola)

parola 5

parola 4

parola 3

parola 2

Parola 1

Alla mia famiglia

ai miei amici e a chi mi è stato accanto.
Che questo traguardo sia solo l'inizio.

Abstract

Abstract qui (ti consiglio di farlo alla fine)

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 0

INTRODUZIONE

Al giorno d'oggi, Internet rappresenta una delle infrastrutture più critiche e pervasive della nostra società, in quanto viene utilizzata costantemente in molteplici ambiti della vita quotidiana: dalla comunicazione personale e professionale che ci tiene connessi a livello globale, all'accesso immediato a una quantità crescente di informazioni e contenuti multimediali, fino alla gestione di transazioni economiche, servizi bancari, amministrazione pubblica e logistica internazionale.

Semplificando, Internet può essere definita come la "rete delle reti": una struttura complessa e gerarchica che consente il collegamento tra milioni di reti locali eterogenee, distribuite in tutto il mondo. A livello architetturale, Internet è composta da un numero elevato, ma finito, di Autonomous System (AS), ciascuno dei quali corrisponde a una rete di proprietà e gestione unificata — come ad esempio un Internet Service Provider (ISP), un'azienda, un'università o un ente governativo — caratterizzata da una propria politica di routing indipendente.
(Si stima l'esistenza di più di 90.000 AS)

Per "politica di routing" si intende l'insieme di regole, preferenze e vincoli che determinano in che modo il traffico di rete viene instradato all'interno dell'AS e verso gli altri sistemi autonomi. Il protocollo di routing incaricato di gestire la propagazione delle informazioni tra AS è il Border Gateway Protocol (BGP), attualmente considerato lo standard de facto per l'interconnessione a livello globale. Il suo com-

pito è annunciare e apprendere rotte [1] , determinando così i percorsi lungo i quali i pacchetti viaggiano da un'estremità all'altra del mondo.

Nonostante la sua centralità e longevità, il protocollo BGP presenta una serie di limiti strutturali, legati al fatto che fu progettato in un'epoca in cui la sicurezza informatica non era ancora una priorità. BGP si basa infatti su un modello di fiducia implicita tra gli operatori di rete e non prevede, nella sua implementazione standard, meccanismi di autenticazione, integrità o verifica delle informazioni propagate. Questa mancanza di sicurezza nativa rende il protocollo vulnerabile a diversi tipi di attacco, tra cui il *prefix hijacking*, il *route leaking* e il *session hijacking*, che possono compromettere seriamente l'affidabilità e la sicurezza della rete Internet.

Alla luce di queste considerazioni, risulta fondamentale analizzare in dettaglio il funzionamento di BGP e le sue vulnerabilità, al fine di individuare le possibili contromisure per prevenire o limitare gli effetti di un eventuale attacco. In un mondo sempre più dipendente dall'utilizzo di Internet, garantire la resilienza e la sicurezza del protocollo di routing interdominio è una priorità non solo tecnica, ma anche strategica e geopolitica.

La tesi si compone di cinque capitoli suddivisi come segue:

- **Primo Capitolo - Autonomous System** Nel primo capitolo viene approfondito il ruolo dell'AS, le sue caratteristiche, a cosa serve, come vengono classificati e altre informazioni utili a comprendere il funzionamento del protocollo BGP.
- **Secondo Capitolo - Protocollo di routing BGP 4.0** In questo capitolo viene analizzato in dettaglio il funzionamento del protocollo BGP (Border Gateway Protocol), attualmente lo standard principale per il routing tra sistemi autonomi su Internet. Dopo una panoramica introduttiva sui concetti fondamentali di routing, viene descritto il contesto storico e tecnico che ha portato all'introduzione di BGP, in particolare come evoluzione del precedente Exterior Gateway Protocol (EGP). Viene poi approfondita la natura del protocollo, la sua collocazione nei livelli del modello di rete e le dipendenze da altri protocolli sottostanti. Una sezione centrale del capitolo è dedicata

[1] Una rotta è il percorso scelto dal protocollo di routing per raggiungere una rete specifica.

al funzionamento interno di BGP: vengono spiegati il meccanismo di routing basato su path vector, l'applicazione delle politiche di importazione ed esportazione delle rotte, gli attributi utilizzati per determinare il miglior percorso e le strategie per evitare la formazione di cicli. Il capitolo si conclude con una descrizione del formato dei messaggi BGP e delle sessioni di peering tra router (iBGP ed eBGP), per poi presentare un confronto tra le versioni storiche del protocollo e le novità introdotte nell'attuale versione 4.0.

- **Terzo Capitolo - Metodologie di attacco e prevenzione del protocollo BGP** Nel terzo capitolo vengono approfondite le principali vulnerabilità del protocollo BGP nella sua ultima versione 4.0, ne vengono analizzate le cause, le modalità di esecuzione dell'attacco e le possibili conseguenze. Vengono trattati tre scenari d'attacco particolarmente rilevanti: il *prefix hijacking*, il *route leaking* e il *session hijacking*. Successivamente, il capitolo introduce due tecnologie progettate per aumentare la sicurezza di BGP: Border Gateway Protocol Security (BGPSec) e Resource Public Key Infrastructure (RPKI). Nella parte finale, viene trattato il tema del monitoraggio delle anomalie BGP attraverso piattaforme pubbliche come *BGPStream* (di Cooperative Association for Internet Data Analysis (CAIDA)), *RIPE RIS* e *Route Views*.
- **Quarto Capitolo - Implicazioni degli attacchi BGP nel mondo** Questo capitolo analizza due noti casi reali di attacchi al protocollo BGP, con l'obiettivo di mostrare le conseguenze concrete che vulnerabilità teoriche possono avere su scala globale: l'BGP hijacking di Pakistan Telecom (2008) e gli episodi attribuiti a China Telecom (2010).
- **Quinto Capitolo - Prospettive future: SDN e BGP** In questo capitolo si discutono le prospettive evolutive dell'interdominio alla luce dei paradigmi Software Defined Networking (SDN), con attenzione a modelli di integrazione, possibili benefici e impatti sulla sicurezza e sulla gestione globale.

Capitolo 1

Autonomous System

In questo capitolo, andiamo ad analizzare cos'è un AS, il suo ruolo nel routing globale e la loro classificazione.

1.1 Il ruolo dell'AS in Internet

Un AS è definito come un insieme di indirizzi IP e router sotto il controllo di una singola entità amministrativa, che adotta una politica id routing uniforme e coerente verso l'esterno. Secondo l'Request For Comment (RFC) 1930 dell'Internet Engineering Task Force (IETF), un AS è necessario ogniqualvolta un'organizzazione desideri definire delle regole di instradamento proprie e differenziate rispetto ad altri domini di routing, oppure quando intrattiene relazioni di peering con più fornitori di connettività a Internet, spesso attraverso gli Internet Exchange Point (IXP). Gli IXP sono infrastrutture fisiche dove diversi AS si connettono per scambiare traffico di rete tra loro in modo diretto. Questa interconnessione diretta, nota come peering, permette agli AS di scambiarsi dati senza dover passare per le reti di transito di terze parti, riducendo i costi e migliorando la latenza. Ogni AS è identificato univocamente dal Autonomous System Number (ASN), assegnato da uno dei cinque Regional Internet Registry (RIR). [1] [5]

1.1.1 Cos'è il Regional Internet Registry

I RIR sono organizzazioni responsabili dell'assegnazione e della registrazione delle risorse numeriche di Internet all'interno di specifiche regioni geografiche del mondo. Le risorse che i RIR assegnano e registrano sono:

- Indirizzi IP sia IPv4 che IPv6.
- ASN usati per identificare gli AS.

Le organizzazioni (come gli ISP, grandi aziende, università, enti governativi, ecc.) che desiderano connettersi a Internet in modo indipendente (ovvero, operare il proprio AS) e/o avere un blocco di indirizzi IP pubblico da gestire direttamente, devono richiedere queste risorse al RIR competente per la loro area geografica.

Attualmente, esistono 5 RIR a livello globale:

- AfriNIC (Africa)
- ARIN (Nord America)
- LACNIC (America Latina e Caraibi)
- APNIC (Asia e Oceania)
- RIPE NCC (Europa, Medio Oriente e parti dell'Asia Centrale)

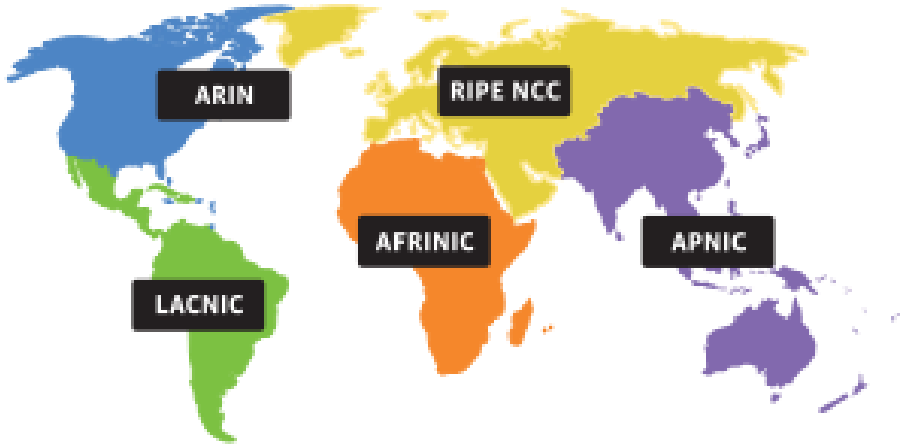

Figura 1.1: Mappa dei RIR[7]

Queste 5 RIR collaborano attraverso l'Number Resource Organization (NRO) e sono sotto la supervisione generale dell'ente Internet Assigned Numbers Authority (IANA), che è il coordinatore globale delle risorse numeriche e dei nomi di dominio di Internet.

1.1.2 Struttura ASN

La nomenclatura degli ASN segue due formati principali: il tradizionale a 16 bit e quello a 32 bit, introdotto successivamente per rispondere all'aumento della domanda (RFC 4893). Gli ASN a 16 bit vanno da 1 a 65535, con alcune riserve speciali: per esempio, i numeri da 64512 a 65534 sono destinati all'uso privato, mentre l'ASN 23456 viene usato come placeholder nella transizione tra 16 e 32 bit (RFC 6996). Il nuovo spazio a 32 bit estende la numerazione fino a 4.294.967.295 e consente anche l'utilizzo della notazione m:n (es. 1:10), che rappresenta una forma più leggibile del numero intero.

Gli ASN vengono spesso preceduti dal prefisso AS (es: AS13335 per Cloudflare, AS15169 per Google) e sono registrati pubblicamente in database come il WHOIS. [1] [19]

1.2 Classificazione AS

Gli Autonomous System possono essere classificati secondo diversi criteri, in base al loro ruolo funzionale nella topologia globale di Internet, alla natura delle connessioni che intrattengono con altri AS, oppure alle politiche di routing adottate.

Una classificazione comunemente diffusa è quella che distingue gli AS in tre grandi categorie:

- **Stub AS**: AS connesso ad un solo provider (single-homed), che non fornisce connettività ad altri AS. È il caso tipico di una rete aziendale o universitaria.
- **Multihomed AS**: AS connesso a più provider, senza però fornire transito ad altri.

[1] WHOIS è un archivio pubblico che raccoglie informazioni sulla titolarità e sull'assegnazione di risors e di rete.

- **Transit AS**: AS che offre connettività ad altri sistemi autonomi, permettendo loro di scambiare traffico. Gli ISP operano tipicamente come transit AS.

Un'altra classificazione si basa sul ruolo gerarchico ricoperto all'interno dell'ecosistema di Internet:

- **Tier 1**: AS che può raggiungere qualsiasi altra rete senza dover acquistare transito da altri AS. Hanno accordi di peering reciproci con gli altri tier di livello 1.
- **Tier 2**: AS che acquista transito (da tier 1), ma può anche offrire servizi a clienti e stabilire peering (offre servizi a tier 3 e fa da peering con altri tier 2).
- **Tier 3**: AS che acquista connettività esclusivamente da altri provider e non fornisce servizi di transito.

Infine, secondo le linee guida dell'IETF (RFC 1930), un Autonomous System è definito anche in base all'autonomia decisionale rispetto alle politiche di routing. Questa indipendenza costituisce uno dei principali motivi per cui un'organizzazione può richiedere un ASN.

Capitolo 2

Protocollo di routing BGP 4.0

Con questo capitolo capiremo tutto sul funzionamento e sulle caratteristiche del protocollo di routing esterno BGP.

2.1 Cos'è il routing

Il routing è il processo attraverso il quale i router determinano il percorso migliore per inviare pacchetti IP da un host sorgente a uno destinatario. Esistono 2 tipi principali di routing:

- Routing interno (Interior Gateway Protocol (IGP)): gestisce il traffico di rete all'interno di uno stesso AS.
- Routing esterno (EGP): gestisce lo scambio di informazioni tra AS diversi.

BGP è un protocollo di routing EGP di tipo *path-vector*, in cui ogni informazione veicolata include il percorso completo di AS attraversati, utile per evitare loop e applicare regole di policy di instradamento.

2.2 Nascita del protocollo BGP

Il protocollo BGP fu sviluppato alla fine degli anni '80 per superare i limiti del precedente protocollo di routing di tipo EGP. La prima versione: BGP-1, fu specificata

tramite RFC 1105[24] nel 1989 da Yakov Rekhter (IBM) e Kirk Lougheed (Cisco) e venne implementata già su router Cisco e il backbone NSFNET.[2] BGP-1 introdusse il concetto di protocollo *path vector*.

Versioni intermedie: BGP-2 e BGP-3

Con l'evoluzione di Internet e l'espansione del backbone NSFNET, fu necessario migliorare le funzionalità iniziali di BGP-1.

- **BGP-2** fu descritto nell'RFC 1163[12] (1990). Questa versione migliorava la gestione delle sessioni TCP tra i peer BGP, raffinando il meccanismo di trasporto dei messaggi e introducendo una maggiore stabilità nella selezione del percorso.
- **BGP-3** arrivò nel 1991 con l'RFC 1267[13]. In questa versione si consolidarono le basi del protocollo, introducendo formalmente i messaggi UPDATE, NOTIFICATION, KEEPALIVE e OPEN nel formato che verrà poi mantenuto anche nella versione successiva. BGP-3 fu largamente utilizzato fino all'introduzione del Classless Inter-Domain Routing (CIDR).

Il salto a BGP-4: CIDR e scalabilità

La vera rivoluzione arrivò con BGP-4, introdotto inizialmente nell'RFC 1654[22] (1994), poi aggiornato da RFC 1771 (1995), e infine formalizzato nell'attuale RFC 4271[23] (2006).

La caratteristica distintiva di BGP-4 fu l'introduzione del supporto al CIDR, che permetteva di:

- Aggregare efficacemente i prefissi IP,
- Ottenere una drastica riduzione delle dimensioni delle tabelle di routing globali,
- Assicurare la scalabilità necessaria per Internet, contrastando la crisi imminente degli indirizzi IPv4.

Non solo, BGP-4 offrì anche un meccanismo robusto per la gestione di attributi di routing personalizzati (come LOCAL_PREF, MED, AS_PATH), il che diede agli operatori la capacità di attuare un routing basato su policy su scala mondiale.

2.2.1 Sostituzione di EGP

Prima dell'introduzione di BGP, l'unico protocollo esterno per l'instradamento tra AS era l'**EGP (Exterior Gateway Protocol)**, formalizzato nella RFC 904[27] (1984). EGP era un protocollo di tipo *distance-vector* molto semplice, progettato per operare in una struttura gerarchica a "stella" centrata sull'AS 1 (Autonomous System principale dell'ARPANET/NSFNET). Esso permetteva esclusivamente di segnalare la raggiungibilità delle reti, senza fornire informazioni sui percorsi o sulle politiche da adottare per instradare il traffico.

Con l'espansione di Internet e l'adozione di topologie a maglia, EGP rivelò rapidamente i suoi limiti:

- **Rigidità topologica**: incapace di operare in reti con interconnessioni multiple tra AS diversi.
- **Assenza di path information**: veniva indicata solo la raggiungibilità, senza dettagli sul percorso.
- **Scarsa scalabilità**: non supportava CIDR, né aggregazione dei prefissi IP.

In risposta a queste esigenze, nel 1989 fu introdotto BGP-1, che adottava un meccanismo path-vector con l'attributo `AS-PATH` per evitare loop e consentire l'applicazione di policy amministrative.

Il passaggio da EGP a BGP fu graduale tra il 1989 e il 1995, periodo durante il quale i principali backbone — incluso NSFNET — migrarono verso BGP-3 e poi BGP-4.

2.3 Caratteristiche protocollo BGP

In questa sezione andiamo a vedere alcune caratteristiche del protocollo BGP, che tipo di protocollo è, dove opera e in che modalità, per poi nella sezione successiva

analizzare effettivamente come avviene il suo funzionamento. Per comprendere appieno le caratteristiche del protocollo BGP è utile richiamare i principi dei modelli di riferimento ISO/OSI e TCP/IP, che ne costituiscono il contesto architetturale.

2.3.1 Modello ISO/OSI e TCP/IP

La progettazione dei protocolli di rete si basa su una struttura a strati, concettualizzata nei modelli di riferimento **ISO/OSI** e **TCP/IP**. Ogni livello di questi modelli svolge un compito specifico e si interfaccia direttamente con i livelli adiacenti, attraverso un meccanismo di scambio di dati chiamato *incapsulamento*.

Durante la trasmissione, ogni volta che un messaggio attraversa un livello verso il basso, viene arricchito con un *header* (intestazione) contenente le informazioni necessarie per il funzionamento del protocollo di quel livello. L'unica eccezione è rappresentata dal primo e dall'ultimo livello. In fase di ricezione, il messaggio risale la pila protocollare: ogni livello rimuove il proprio header e interpreta i dati contenuti, secondo la logica del *decapsulamento*.

Ogni livello può supportare più protocolli, ma ciascun protocollo appartiene esclusivamente al livello in cui opera. L'**entità** è l'unità funzionale all'interno di un livello che implementa uno specifico protocollo. Di conseguenza, il numero di entità presenti in un livello coincide con il numero di protocolli gestiti. Durante la comunicazione, è sempre una sola entità per livello ad aggiungere o rimuovere l'header del proprio protocollo.

Le entità omologhe situate agli stessi livelli dei due nodi comunicanti prendono il nome di *peer entities*, mentre i livelli stessi sono detti *peer levels*. Le informazioni scambiate tra livelli adiacenti sono chiamate Protocol Data Unit (PDU), che includono i dati e l'eventuale header del livello corrente.

Ogni entità è identificata univocamente da un indirizzo detto Service Access Point (SAP), che consente la comunicazione tra livelli differenti.

I protocolli di rete si distinguono inoltre in:

- **Connectionless**, che privilegiano la velocità di trasmissione e non instaurano un canale persistente (es. UDP);

- **Connection-oriented**, che instaurano una connessione stabile per garantire affidabilità (es. TCP).

Dal punto di vista architetturale, il modello **TCP/IP** rappresenta lo standard *de facto*, utilizzato nella rete Internet fin dalle sue origini. Il modello **ISO/OSI**, invece, costituisce lo standard *de jure* proposto dall'ISO.[3]

La struttura e la corrispondenza tra i due modelli sono illustrate nella figura seguente, che evidenzia i livelli funzionali e la loro equivalenza logica:

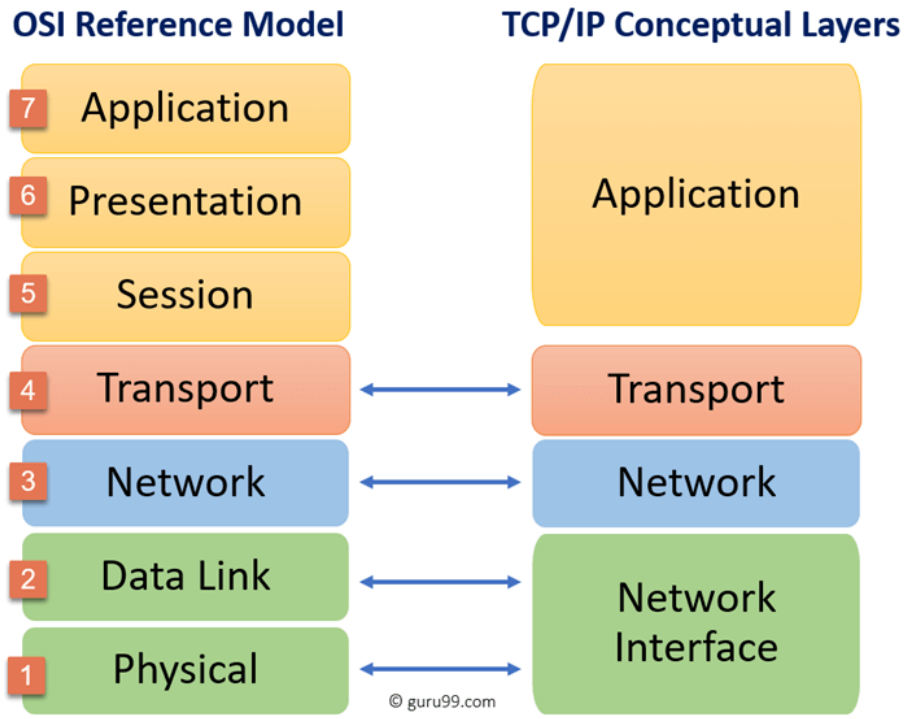

Figura 2.1: Modello ISO/OSI e modello TCP/IP

Di seguito si riporta una breve descrizione di cosa si occupa ciascun livello:

- **Physical**: definisce le caratteristiche fisiche del mezzo di trasmissione (cavi, segnali elettrici/ottici, connettori) e la codifica dei bit.
- **Data Link**: gestisce l'incapsulamento dei dati in frame, il controllo degli errori a livello di collegamento e l'indirizzamento fisico (MAC address).

- **Network**: si occupa dell'instradamento dei pacchetti tra reti differenti, includendo il calcolo del percorso e l'indirizzamento logico (es. indirizzi IP).
- **Transport**: garantisce il trasferimento dei dati end-to-end, controllando affidabilità, segmentazione, riordinamento e gestione della congestione (es. TCP e UDP).
- **Session** (solo modello OSI): gestisce l'instaurazione, il mantenimento e la terminazione delle sessioni di comunicazione tra applicazioni.
- **Presentation** (solo modello OSI): si occupa della sintassi e della semantica delle informazioni trasmesse, includendo cifratura, compressione e traduzione dei dati.
- **Application**: fornisce servizi di rete direttamente alle applicazioni utente (es. HTTP, FTP, SMTP, DNS).

Per quanto riguarda i dispositivi fisici associati ai vari livelli:

- La **scheda di rete** (NIC) è un apparato di livello 1 (Physical).
- Uno **switch** opera a livello 2 (Data Link).
- Un **router** lavora a livello 3 (Network).
- Un **host**, come un computer personale, implementa tutti i livelli, fino al livello 7 (Application) nel modello OSI.

2.3.2 Collocazione architetturale e modalità operative di BGP

Dopo aver analizzato i modelli di riferimento ISO/OSI e TCP/IP, è possibile inquadrare il BGP nel contesto architetturale delle reti e descriverne le principali caratteristiche operative. BGP è un protocollo di routing esterno di tipo *path-vector*, progettato per lo scambio di informazioni di raggiungibilità tra AS differenti. Opera al livello **Application** sia nel modello ISO/OSI sia in quello TCP/IP e utilizza una modalità di comunicazione **connection-oriented**, basata su una connessione **TCP** stabile (porta 179) per garantire affidabilità, ordinamento e controllo degli errori

nella consegna dei messaggi. A differenza dei protocolli di routing interno, BGP applica criteri di instradamento non solo tecnici (costo, distanza, latenza, banda), ma anche amministrativi (le policy), consentendo agli amministratori di un AS di stabilire con precisione quali rotte accettare, rifiutare o preferire, in base a politiche economiche, di sicurezza o di accordi di peering. [16]

2.3.3 Il ruolo del router e le tabelle di routing

Il router è il dispositivo di rete incaricato di instradare i pacchetti IP tra reti differenti. Per svolgere questa funzione, mantiene al suo interno strutture dati chiamate **tabelle di routing**, che contengono le informazioni necessarie a determinare il *next hop*, ossia il prossimo nodo a cui inoltrare un pacchetto destinato a una certa rete.

Una voce tipica di tabella di routing comprende:

- l' **IP di rete** (es. 192.168.0.0/24), che identifica l'insieme di indirizzi raggiungibili;
- la **subnet mask**, che specifica la dimensione del prefisso;
- il **next hop**, ovvero l'indirizzo IP del router di transito successivo;
- l'**interfaccia di uscita** locale, usata per inoltrare il pacchetto verso il next hop;
- eventuali **metriche o attributi** che descrivono la qualità o la preferenza della rotta.

È importante distinguere tra due concetti:

- **Tabella di routing** (*Routing Information Base (RIB)*): contiene tutte le rotte conosciute dal router, apprese sia tramite protocolli di routing (es. OSPF, BGP) sia tramite configurazioni statiche.
- **Tabella di forwarding** (*Forward Information Base (FIB)*): è una versione ottimizzata della tabella di routing, mantenuta spesso direttamente nell'hardware per garantire l'inoltro ad alta velocità. Mentre la tabella di routing può contenere più alternative per la stessa destinazione, la FIB conserva solo la rotta effettivamente selezionata (la *best path*).

In questo modo, i protocolli di routing come BGP non instradano i pacchetti direttamente, ma aggiornano le tabelle di routing del router. Sarà poi il meccanismo di forwarding a occuparsi dell'inoltro vero e proprio dei pacchetti, basandosi sulla FIB.

2.3.4 Le tabelle di routing in BGP

Per descrivere in modo chiaro il funzionamento di BGP, è utile distinguere tre insiemi logici di tabelle di routing che vengono fatti dal protocollo:

- **Adj-RIBs-In**: contengono tutte le rotte apprese dai peer BGP tramite i messaggi `UPDATE`, ancora prima che vengano applicati filtri o politiche di selezione;
- **Loc-RIB**: rappresenta la tabella BGP locale, in cui confluiscono le rotte selezionate come "migliori" dal Decision Process;
- **Adj-RIBs-Out**: raccolgono le rotte che, dopo l'applicazione delle policy di esportazione, sono pronte per essere annunciate ai peer.

Queste strutture non corrispondono a implementazioni fisiche obbligatorie nei router, ma costituiscono un modello logico standardizzato, utilizzato per descrivere come le informazioni passano da un peer all'altro. In questo modo è più semplice comprendere su quali insiemi di rotte agiscono il processo decisionale e le politiche di filtro o manipolazione.

2.4 Funzionamento BGP

In questa sezione analizziamo nel dettaglio come il protocollo BGP adempie al suo compito principale: permettere il routing tra i diversi AS, selezionando il percorso più vantaggioso secondo criteri sia tecnici sia amministrativi.[23]

2.4.1 Tipologie di sessione

BGP stabilisce connessioni affidabili tra router tramite **TCP** (porta 179), demandando al livello transport le funzioni di controllo degli errori, ordinamento e ritrasmissione. Le connessioni BGP, chiamate *sessioni*, possono essere di due tipi:

- **eBGP (External BGP)**: instaurate tra router appartenenti a AS differenti. Vengono utilizzate per scambiare informazioni di raggiungibilità IP in ambito interdominio, secondo lo schema CIDR.
- **iBGP (Internal BGP)**: instaurate tra router dello stesso AS. Servono a distribuire internamente le informazioni acquisite dai peer esterni.

2.4.2 Caratteristiche del protocollo path-vector

BGP è un protocollo di routing esterno di tipo **path-vector**, evoluzione dei protocolli distance-vector. Invece di propagare solo una distanza numerica, BGP include in ogni annuncio l'intera sequenza di AS (`AS_PATH`) da attraversare per raggiungere una destinazione.

Questo approccio presenta due vantaggi principali:

- **Prevenzione dei cicli**: se un router riceve un annuncio che contiene già il proprio AS nel campo `AS_PATH`, quell'annuncio viene scartato, evitando loop di instradamento.
- **Supporto alle policy**: la lista di AS permette di applicare criteri non solo tecnici ma anche amministrativi, in base a rapporti economici o di peering.

2.4.3 Policy di routing

BGP è un protocollo **policy-based**: le rotte non vengono accettate o pubblicizzate solo in base a metriche di costo, ma in base a policy locali stabilite dall'amministratore.

- **Export policy**: un AS decide quali reti IP (identificate in notazione CIDR) rendere note ai vicini. Ad esempio, una rete aziendale (Stub AS) può annunciare solo i propri prefissi interni senza offrire transito.
- **Import policy**: un AS può rifiutare rotte che passano da sistemi indesiderati (es. per motivi di sicurezza o commerciali) oppure preferire rotte provenienti da determinati peer.

Questo rende BGP molto flessibile ma introduce anche maggiore complessità nella gestione delle tabelle di routing e un maggior consumo di memoria nei router, poiché è necessario mantenere molteplici informazioni di percorso.

2.4.4 Formato dei messaggi

Tutti i messaggi BGP hanno una struttura comune, composta da:

- **Marker** (16 byte): usato originariamente per autenticazione, oggi spesso a valore fisso.
- **Length** (2 byte): indica la lunghezza totale del messaggio (da 19 a 4096 byte).
- **Type** (1 byte): specifica il tipo di messaggio, tra i seguenti:

1. **OPEN**: il primo messaggio scambiato, utilizzato per instaurare la sessione. Contiene numero di AS, Hold Time, BGP Identifier ed eventuali parametri opzionali.
2. **UPDATE**: trasporta annunci e ritiri di Network Layer Reachability Information (NLRI) (ovvero le informazioni di raggiungibilità delle reti IP identificate da prefissi in notazione CIDR) insieme ai relativi attributi di percorso.
3. **NOTIFICATION**: segnala errori e causa l'immediata chiusura della sessione.
4. **KEEPALIVE**: inviato periodicamente per confermare la validità della sessione in assenza di aggiornamenti.

2.4.5 Gestione delle sessioni (FSM)

L'instaurazione e la gestione di una sessione BGP sono regolate da una **macchina a stati finiti** (Finite State Machine, FSM) definita nell'RFC 4271. Essa descrive il comportamento di un router BGP in risposta a eventi interni (scadenza di timer) o esterni (ricezione di messaggi, esito delle connessioni TCP).

Gli stati principali sono sei:

- **Idle**: Stato iniziale, in cui il router non ha ancora connessioni attive. La sessione BGP può essere avviata:

  - tramite *Manual Start*, ossia un comando esplicito dell'amministratore di rete;
  - tramite *Automatic Start*, cioè l'avvio automatico del processo BGP, ad esempio al riavvio del router.

  In entrambi i casi, il router inizializza le risorse necessarie alla sessione: azzera i contatori (*ConnectRetryCounter*), attiva i timer (*ConnectRetryTimer*) e alloca le strutture di controllo interne. Successivamente:

  - se configurato come **peer attivo**, tenta di instaurare una connessione TCP verso il vicino e passa allo stato **Connect**;
  - se configurato come **peer passivo**, rimane in ascolto di connessioni in ingresso e passa allo stato **Active**.

- **Connect**: Il router ha avviato un tentativo di connessione TCP verso il peer e attende il completamento del 3-way handshake. Se la connessione fallisce o scade il *ConnectRetryTimer*, viene avviato un nuovo tentativo: a seconda della configurazione e degli eventi, il router può rimanere in **Connect** o transitare in **Active**. In caso di successo della connessione TCP, si procede con l'invio del messaggio *OPEN* (se l'opzione *DelayOpen* è disabilitata) oppure con l'attesa dello scadere del *DelayOpenTimer*.

- **Active**: Stato in cui il router non ha ancora stabilito una connessione TCP valida. In questa fase rimane in ascolto di connessioni in ingresso dal peer. Se entro lo scadere del *ConnectRetryTimer* non arriva alcuna connessione, il router torna in **Connect** per avviare un nuovo tentativo attivo di connessione.

- **OpenSent**: in questo stato il router ha inviato un messaggio *OPEN* e attende l'*OPEN* dal peer. Se il messaggio ricevuto è valido, vengono negoziati i parametri della sessione (in particolare l'*HoldTimer* e il *BGP Identifier*) e si passa a *OpenConfirm*. In caso di errore o incompatibilità, viene inviato un *NOTIFICATION* e la sessione viene chiusa.

- **OpenConfirm**: i peer attendono un *KEEPALIVE* di conferma. Se il messaggio arriva entro l'*HoldTimer*, si passa allo stato finale *Established*. Se invece il

timer scade prima, viene inviato un *NOTIFICATION* con codice *Hold Timer Expired*.

- **Established**: la sessione è attiva e operativa. In questo stato vengono scambiati messaggi *UPDATE*, *KEEPALIVE* e *NOTIFICATION*. La ricezione di *KEEPALIVE* o *UPDATE* resetta l'*HoldTimer*, mantenendo viva la connessione. Qualsiasi errore o scadenza dei timer comporta la chiusura della sessione e il ritorno allo stato *Idle*.

Questa struttura a stati consente a BGP di garantire un comportamento prevedibile e robusto in ogni fase della connessione, riducendo ambiguità e rendendo il protocollo resistente a errori e disconnessioni temporanee.

2.4.6 Path attributes

Ogni annuncio BGP (UPDATE) include attributi che specificano le proprietà della rotta. Essi si classificano in:

- **Well-known mandatory**: sempre presenti (es. `ORIGIN`, `AS_PATH`, `NEXT_HOP`);
- **Well-known discretionary**: riconosciuti da tutti ma non obbligatori (es. `LOCAL_PREF`);
- **Optional transitive**: opzionali, ma se non riconosciuti vengono comunque propagati (es. `COMMUNITY`);
- **Optional non-transitive**: opzionali e non propagati se non riconosciuti (es. `MED`);
- **Partial**: optional-transitive che non sono stati riconosciuti da un router e vengono inoltrati con un marcatore speciale.

Tra gli attributi più importanti:

Attributo	Classificazione	Descrizione
ORIGIN	Well-known mandatory	Specifica l'origine della rotta: *IGP* se appresa internamente, *EGP* se tramite un protocollo esterno, *INCOMPLETE* se l'origine non è determinata (es. redistribuzione da altre fonti).
AS_PATH	Well-known mandatory	Elenca in sequenza gli Autonomous System attraversati dalla rotta. È usato sia per la prevenzione dei loop (se un AS vede se stesso nella lista, scarta l'update) sia come criterio di selezione preferendo i percorsi con meno AS.
NEXT_HOP	Well-known mandatory	Indirizzo IP del router BGP da utilizzare come prossimo destinatario per raggiungere la destinazione finale. Determina concretamente a chi inoltrare il traffico.
LOCAL_PREF	Well-known discretionary	Valore propagato solo all'interno dello stesso AS (iBGP). Indica la preferenza per le rotte in uscita: percorsi con valore più alto sono scelti come prioritari.
ATOMIC_-AGGREGATE	Well-known discretionary	Segnala che la rotta è stata ottenuta tramite un processo di aggregazione, e quindi alcune informazioni più specifiche (es. dettagli di prefissi originari) possono essere andate perse.
MED	Optional non-transitive	Multi Exit Discriminator: viene usato per suggerire quale collegamento preferire quando esistono più punti di ingresso verso lo stesso AS adiacente. Valori più bassi sono considerati migliori, ma l'attributo non è sempre rispettato da tutti i peer.
COMMUNITY	Optional transitive	Etichette logiche che consentono di raggruppare insiemi di rotte e applicare politiche comuni (es. blocco, preferenza, redistribuzione). Sono molto usate per semplificare la gestione del routing.
AGGREGATOR	Optional transitive	Identifica l'AS e il router che hanno effettuato un'aggregazione di rotte. È utile per tracciare l'origine del processo di aggregazione e garantire trasparenza nell'instradamento.

Tabella 2.1: Principali attributi BGP e loro classificazione

2.4.7 Processo

Il **Decision Process** costituisce il cuore di BGP ed è responsabile della selezione della rotta "migliore" verso una determinata destinazione, quando ne esistono più di una. A differenza dei protocolli interni (IGP), che si basano su metriche uniche come costo o latenza, BGP adotta un approccio multilivello, in cui criteri amministrativi e tecnici sono combinati in modo gerarchico.[23]

Il processo decisionale si applica a ogni informazione di raggiungibilità (NLRI) ricevuta tramite messaggi `UPDATE`. Quando un router BGP riceve per la prima volta un UPDATE da peer esterni o interni (quindi da router con cui non aveva ancora una conoscenza diretta delle reti raggiungibili) inserisce tutte le rotte candidate nella propria *Adj-RIBs-In*. Da questo insieme viene poi selezionata una sola rotta preferita, che sarà utilizzata per l'instradamento effettivo e potenzialmente annunciata ad altri peer.

Il processo si articola in tre fasi principali:

1. **Assegnazione della preferenza locale**: Ogni rotta viene inizialmente valutata in base a criteri interni all'AS. In questa fase è fondamentale l'attributo `LOCAL_PREF`, che consente all'amministratore di indicare quale uscita preferire in modo indipendente dai parametri tecnici. Questo garantisce che, anche in presenza di molte alternative, la scelta rifletta le politiche di instradamento locali (es. preferire collegamenti verso clienti anziché verso peer o provider).
2. **Selezione della rotta migliore**: Dopo l'applicazione delle policy, le rotte rimaste vengono confrontate seguendo un ordine di priorità ben definito dallo standard. In particolare:
   (a) scegliere la rotta con **LOCAL_PREF** più alto;
   (b) a parità, preferire quella con **AS_PATH più corto**;
   (c) a parità, preferire il valore di **ORIGIN** più favorevole (IGP < EGP < INCOMPLETE);
   (d) se le rotte provengono dallo stesso AS adiacente, scegliere quella con **MED** più basso;

(e) preferire rotte apprese tramite **eBGP** rispetto a quelle iBGP;

(f) scegliere la rotta il cui `NEXT_HOP` è raggiungibile con il **costo IGP minore**;

(g) in caso di ulteriore parità, selezionare la rotta proveniente dal router con **BGP Identifier** più basso;

(h) come criterio finale, utilizzare l'indirizzo IP più basso del peer.

Questo meccanismo garantisce che, anche quando più router sconosciuti iniziano a scambiarsi informazioni, ognuno di essi arrivi a determinare una sola rotta coerente e stabile per ogni destinazione.

3. **Disseminazione della rotta**: La rotta vincente viene installata nella *Loc-RIB* (tabella BGP locale). In base alle export policies, può poi essere inserita nelle *Adj-RIBs-Out* e annunciata ai peer tramite messaggi `UPDATE`. In questa fase possono essere applicate tecniche di manipolazione come l'aggregazione dei prefissi IP o l'*AS_PATH prepending*, strumenti utili per influenzare le scelte di instradamento dei router vicini.

In questo modo, BGP non opera come un classico protocollo di "shortest path", ma come un sistema flessibile e distribuito di negoziazione delle rotte, in cui le politiche commerciali (cliente, provider, peer) contano quanto i parametri tecnici. La convergenza emerge dal fatto che ogni router, anche senza conoscenze pregresse della topologia globale, applica localmente lo stesso processo decisionale definito dallo standard, ottenendo così una rete interdominio coerente e priva di cicli.

2.4.8 Considerazioni finali

Grazie alla combinazione di meccanismi tecnici (AS_PATH, NEXT_HOP, ORIGIN), controlli di robustezza (FSM, HoldTimer, NOTIFICATION) e criteri amministrativi (import/export policies), BGP rappresenta il cuore del routing interdominio su Internet. È un protocollo pensato non per la rapidità di convergenza, ma per la stabilità e la possibilità di esprimere politiche complesse, che riflettono le relazioni economiche e tecniche tra i numerosi AS.

Capitolo 3

Metodologie di attacco e prevenzione del protocollo BGP

In questo capitolo andiamo a vedere quali vulnerabilità del protocollo BGP possono essere sfruttate, come e soprattutto come prevenire che ciò avvenga.

3.1 BGP prefix hijacking

Il **prefix hijacking** è considerato una delle vulnerabilità più gravi del protocollo BGP. In letteratura viene definito come l'annuncio, da parte di un Autonomous System, di prefissi IP che non gli appartengono. Poiché BGP non prevede nativamente meccanismi di autenticazione, i router che ricevono l'annuncio non sono in grado di verificare se l'AS originator sia realmente autorizzato, e quindi possono accettare e propagare l'informazione falsa.[32].

3.1.1 Descrizione dell'attacco.

Nel funzionamento ordinario, un router BGP annuncia ai propri peer i prefissi IP che può raggiungere, insieme agli attributi di percorso come `AS_PATH`, `NEXT_HOP` e altri parametri. Questi annunci vengono propagati e installati nelle tabelle *Adj-RIBs-In* dei router vicini, che li valutano secondo il proprio processo decisionale.

Nel caso di prefix hijacking, un AS trasmette intenzionalmente (o per un errore di configurazione) un `UPDATE` che dichiara la raggiungibilità di un prefisso IP non assegnato a sé. Poiché BGP non integra meccanismi di autenticazione dell'origine, gli AS riceventi non hanno modo di distinguere un annuncio legittimo da uno malevolo. Se l'annuncio falso risulta più vantaggioso rispetto a quello corretto (ad esempio per un `AS_PATH` più corto o per la presenza di un sottoprefisso) la rotta hijacked può venire selezionata come *best path* ed entrare in propagazione globale. In questo modo, il traffico destinato al prefisso dirottato viene instradato attraverso l'AS attaccante, che può semplicemente scartarlo, analizzarlo o deviarlo altrove.

3.1.2 La regola del Longest Prefix Match

Per comprendere il *Subprefix hijack* è necessario richiamare la regola del **Longest Prefix Match (LPM)**. Un router, quando deve decidere come instradare un pacchetto IP, sceglie la rotta più specifica nella propria tabella di routing, ossia quella con la subnet mask più alta.

- La rotta `192.0.2.0/23` copre 512 indirizzi (da `192.0.2.0` a `192.0.3.255`).
- La rotta `192.0.2.0/24` copre solo 256 indirizzi (da `192.0.2.0` a `192.0.2.255`), quindi è più specifica.

Se nella tabella sono presenti entrambe, il router sceglierà sempre `/24` per un pacchetto destinato a `192.0.2.55`, anche se `/23` include comunque quell'indirizzo. Questa proprietà, normalmente utile per ottimizzare l'instradamento, diventa l'elemento chiave sfruttato negli attacchi di tipo Subprefix hijack.

3.1.3 Tipologie di BGP Prefix Hijacking

Si distinguono due tipologie principali di hijack:

- **Regular prefix hijack**: l'AS attaccante annuncia lo stesso prefisso dell'AS legittimo, presentandosi come origin AS. La propagazione del falso annuncio provoca una parziale deviazione del traffico, dipendente dalle politiche di routing adottate dai diversi AS.

Esempio pratico: Immaginiamo di essere un AS, l'AS100, che possiede il blocco IP `203.0.113.0/24`. Normalmente, i suoi router annunciano questo prefisso ai peer BGP, così il mondo intero sa che per raggiungere `203.0.113.x` deve passare da AS100. Ora entra in gioco un attaccante, AS666, che anche se non possiede il blocco IP `203.0.113.0/24`, può inviare un `UPDATE` in cui annuncia di raggiungerlo. Se il percorso verso AS666 è considerato più conveniente (per esempio grazie a un `AS_PATH` più corto), alcuni peer inizieranno a instradare il traffico verso di lui, causando una deviazione parziale.

- **Subprefix hijack**: l'attaccante annuncia un prefisso più specifico rispetto a quello legittimo (ad esempio, `192.0.2.0/24` invece di `192.0.2.0/23`). A causa della regola del *Longest Prefix Match*, quasi tutto il traffico destinato a `192.0.2.x` verrà instradato verso l'attaccante, mentre il legittimo AS continuerà a ricevere solo il traffico diretto a `192.0.3.x`. Questa variante è la più pericolosa, in quanto consente un dirottamento pressoché completo.

3.2 BGP route leaking

La seconda vulnerabilità di BGP che andremo ad analizzare è detta **BGP Route Leaking**. A differenza del *prefix hijacking*, in cui un AS annuncia prefissi non posseduti, nel *route leak* i prefissi sono legittimi ma vengono diffusi in modo improprio, raggiungendo domini di instradamento ai quali non erano destinati. Questa anomalia può deviare grandi quantità di traffico su cammini non ottimali o addirittura inaffidabili, causando disservizi difficili da diagnosticare e potenzialmente con impatto globale[28;6].

3.2.1 Descrizione dell'attacco

Un *BGP route leak* si verifica quando un AS inoltra rotte ricevute da un provider o da un peer verso un altro provider o peer, invece di limitarle ai propri clienti. Questo comportamento viola il principio del *valley-free routing*, che impone che:

- le rotte ricevute da **provider** o **peer** vengano propagate **solo verso clienti**;

- le rotte ricevute da **clienti** possono essere propagate verso qualsiasi altro AS (provider, peer o altri clienti).

In termini economici, questo modello assicura che un AS non faccia involontariamente da punto di transito non compensato.

L'RFC 7908 classifica sette tipologie di route leaks, tra cui le più comuni sono:

- **Provider → Provider**: un AS riceve rotte da un provider e le pubblicizza a un altro provider;
- **Provider → Peer**: un AS inoltra a un peer rotte apprese da un provider;
- **Peer → Peer**: un AS diffonde a un peer rotte ricevute da un altro peer;
- **Peer → Provider**: un AS pubblicizza a un provider rotte che aveva appreso da un altro peer.

3.3 BGP session hijacking

Terzo e ultimo attacco al protocollo che tratteremo è il *BGP session hijacking*. Il BGP session hijacking rappresenta una minaccia che agisce direttamente sulla connessione tra due peer BGP, compromettendo la sessione TCP sottostante e mettendo a rischio la stabilità e la sicurezza del protocollo.[21]

3.3.1 Descrizione dell'attacco

Il *BGP session hijacking* è una delle vulnerabilità più critiche del BGP. Questo tipo di attacco si verifica quando un attore malevolo riesce a prendere il controllo di una sessione TCP già stabilita tra due router BGP, sfruttando la mancanza di meccanismi di autenticazione robusti nel protocollo. Poiché BGP si basa su TCP (porta 179), un attaccante che riesce a predire o intercettare i numeri di sequenza TCP può inserire pacchetti falsi nella connessione, con la possibilità di iniettare o manipolare messaggi BGP legittimi.

3.3.2 Dinamica dell'attacco

L'attacco si articola generalmente in tre fasi principali:

1. **Identificazione della sessione**: l'attaccante individua i due router coinvolti in una connessione BGP attiva.
2. **Predizione o intercettazione dei numeri di sequenza TCP**: attraverso sniffing del traffico o tentativi di brute force, l'attaccante ottiene i valori corretti per inserirsi nella comunicazione.
3. **Iniezione di pacchetti malevoli**: una volta presa la sincronizzazione, l'attaccante può iniettare messaggi BGP alterati, forzando i peer a modificare la loro tabella di routing.

Un attacco di questo tipo può portare a:

- interruzione della connessione BGP tra i router, con conseguente instabilità della rete;
- manipolazione delle rotte, con possibilità di deviare il traffico verso destinazioni non legittime;
- esposizione del traffico intercettato a tecniche di analisi o modifica.

3.4 BGPSec

Il protocollo **BGPsec** è un'estensione del protocollo BGP sviluppata dall'IETF con l'obiettivo di migliorare la sicurezza nella propagazione delle informazioni di routing. A differenza di BGP tradizionale, che si limita a propagare gli annunci senza alcuna forma di autenticazione crittografica, BGPsec introduce un sistema di validazione del percorso basato su firme digitali, permettendo così la verifica crittografica di ogni hop all'interno dell'AS-PATH[11].

3.4.1 Come funziona BGPsec

Dal punto di vista tecnico, BGPsec sostituisce il tradizionale attributo `AS_PATH` con una nuova struttura, chiamata `BGPsec_Path`, che contiene due componenti principali:

- **Secure_Path Segment**: registra l'identità (ASN) di ciascun AS che ha propagato l'annuncio.
- **Signature Segment**: contiene la firma digitale generata da quell'AS sull'intero annuncio ricevuto, calcolata con l'algoritmo Elliptic Curve Digital Signature Algorithm (ECDSA) su curva P-256 e hash SHA-256, come specificato in RFC 8608 [29]

Per un approfondimento teorico e matematico sull'algoritmo ECDSA si rimanda all'Appendice A.

Il flusso operativo è il seguente:

1. L'AS originator di un prefisso genera un messaggio `UPDATE` e lo firma con la propria chiave privata, associata a un certificato valido della RPKI.
2. Quando un AS riceve l'annuncio:
   (a) verifica tutte le firme presenti nei segmenti precedenti utilizzando le chiavi pubbliche distribuite tramite RPKI;
   (b) se l'annuncio è valido, aggiunge il proprio ASN in un nuovo `Secure_Path Segment`;
   (c) calcola e allega la propria firma digitale in un `Signature Segment`.
3. L'annuncio così aggiornato viene propagato al peer successivo, che ripete lo stesso processo.

In questo modo ogni router non solo apprende il percorso di instradamento, ma può anche verificare crittograficamente che:

1. l'annuncio sia stato originato da un AS legittimo (validazione dell'origination);
2. l'AS-PATH non sia stato manomesso lungo il percorso (path validation).

Tale meccanismo riduce drasticamente la possibilità che un attaccante possa inserire ASN falsi o alterare il path, poiché qualsiasi modifica non autorizzata invaliderebbe la catena di firme.

3.4.2 Vulnerabilità risolte

BGPsec nasce come risposta alle principali vulnerabilità storiche di BGP, in particolare:

- **Prefix hijacking**: grazie alla validazione crittografica, diventa più difficile per un AS non autorizzato annunciare prefissi IP di cui non è legittimo proprietario.
- **Session hijacking**: sebbene questo attacco agisca a livello di trasporto (TCP), BGPsec mitiga gli effetti legati alla manipolazione degli annunci, poiché un attaccante non può modificare la catena di firme senza invalidarla.
- **Route leaks e man-in-the-middle**: la verifica delle firme lungo il path riduce la possibilità che un AS introduca rotte false o modifichi l'AS-PATH senza essere rilevato.

Nonostante ciò, BGPsec non elimina tutte le debolezze di BGP: rimangono ad esempio problematiche legate alla disponibilità (DoS) e all'adozione pratica su larga scala.

3.5 BGP RPKI

La **RPKI** è una tecnologia sviluppata dall'IETF per rafforzare la sicurezza del routing interdominio. Il suo obiettivo principale è contrastare attacchi che sfruttano l'assenza di autenticazione in BGP, come il *prefix hijacking*, fornendo un meccanismo crittografico che permette di stabilire quali AS siano legittimati ad annunciare determinati prefissi IP. RPKI rappresenta dunque il fondamento della cosiddetta *origin validation* e costituisce la base crittografica anche per estensioni più avanzate come BGPsec. [8]

3.5.1 Come funziona BGP RPKI

La **RPKI** è un'infrastruttura a chiave pubblica che associa le risorse di rete (prefissi IP e ASN) ai rispettivi detentori legittimi. Ogni allocazione di indirizzi IP e ASN

viene certificata mediante certificati X.509 conformi allo standard PKIX[1], emessi dalle autorità di registrazione regionali (RIR).

Un elemento centrale di RPKI è il *Route Origin Authorization (ROA)*, un oggetto firmato digitalmente che specifica quali ASN sono autorizzati ad annunciare un determinato prefisso. Quando un router riceve un annuncio BGP, può consultare la RPKI per verificare se l'AS originator è autorizzato ad annunciare quel prefisso:

- **Valid**: il prefisso è autorizzato dall'AS specificato nel ROA.
- **Invalid**: l'annuncio non è coerente con i ROA (potenziale hijack).
- **NotFound**: non esiste alcun ROA per il prefisso.

3.5.2 Vulnerabilità risolte

RPKI nasce come soluzione per mitigare attacchi legati all'**origin validation**, in particolare:

- **Prefix hijacking**: impedisce a un AS malevolo di annunciare prefissi IP di cui non possiede i diritti.
- **Route leaks**: riduce la probabilità che un annuncio non autorizzato venga accettato globalmente.

Tuttavia, RPKI non valida l'intero percorso (AS-PATH), quindi non impedisce modifiche intermedie o manipolazioni del path; per questo è considerata complementare a BGPsec.

3.6 Monitoraggio e rilevamento anomalie

Il protocollo BGP, nella sua versione base, è privo di meccanismi di sicurezza intrinseci, e quindi esposto ad attacchi come session hijacking, route leaking e prefix hijacking Per questo motivo, osservare il piano di controllo di Internet, cioè i mecca-

[1]PKIX (*Public Key Infrastructure X.509*) è uno standard IETF che definisce l'uso dei certificati digitali X.509 per costruire infrastrutture a chiave pubblica (PKI). Specifica formati, algoritmi e procedure di validazione per garantire autenticità e integrità nelle comunicazioni sicure.

nismi con cui i router definiscono e aggiornano le rotte, è essenziale per individuare anomalie e preservare l'affidabilità della rete.

Negli ultimi decenni sono stati sviluppati diversi progetti e piattaforme che raccolgono e distribuiscono dati BGP su scala globale, consentendo di:

- rilevare tempestivamente anomalie nella propagazione degli annunci;
- analizzare eventi di sicurezza come attacchi di hijacking o manipolazioni del path;
- studiare l'evoluzione a lungo termine della stabilità del routing interdominio.

Tra i principali strumenti e dataset disponibili si trovano **BGPStream**, il **CAIDA**, il **Routing Information Service (RIS)** di RIPE NCC e il progetto **Route Views** dell'Università dell'Oregon. Ognuno di essi fornisce dati complementari e metodologie differenti, permettendo così un'analisi più completa e accurata delle dinamiche BGP a livello globale.

3.6.1 BGPStream (CAIDA)

BGPStream è una piattaforma open-source sviluppata da CAIDA per l'analisi di grandi dataset BGP. Non fornisce dati propri, ma mette a disposizione un framework software che permette di accedere e analizzare, in tempo reale o in modalità storica, i flussi di messaggi BGP (UPDATE e RIB dumps) provenienti da fonti come RIPE RIS e Route Views. Grazie a questa astrazione, i ricercatori possono integrare e confrontare dataset eterogenei con strumenti uniformi, facilitando il rilevamento di anomalie, lo studio di eventi di hijacking e il monitoraggio della stabilità del piano di controllo globale.[20]

3.6.2 RIPE RIS

Il **RIS** di RIPE NCC stabilisce sessioni BGP con centinaia di router (peer) distribuiti globalmente, chiamati *Route Collectors*. Da queste sessioni vengono generati due insiemi principali di dati:

- **UPDATE streams**: sequenze di messaggi BGP in tempo reale, che mostrano modifiche e anomalie nella propagazione dei prefissi.
- **RIB dumps**: snapshot periodici delle tabelle di routing complete osservate da ciascun collector.

Questi dataset, disponibili pubblicamente dal 2001, costituiscono una risorsa fondamentale per analizzare la diffusione di prefissi, rilevare fenomeni di hijacking e condurre studi longitudinali sul comportamento di BGP.[26]

3.6.3 Route Views

Il progetto **Route Views**, ospitato presso la University of Oregon, opera in maniera analoga a RIPE RIS, mantenendo sessioni BGP con un ampio insieme di Autonomous System a livello globale. I dati resi disponibili comprendono:

- **BGP UPDATE messages**: archiviati con granularità temporale fine, utili per ricostruire eventi di routing e anomalie.
- **RIB snapshots**: dump regolari delle tabelle di routing complete, che forniscono una visione storica dettagliata dell'evoluzione della connettività inter-AS.

Avviato nel 1997, Route Views rappresenta uno dei dataset storici più longevi e utilizzati per studi accademici e per la rilevazione di anomalie di instradamento su Internet.[30]

Capitolo 4

Implicazioni degli attacchi BGP nel mondo

Gli episodi di hijacking hanno mostrato come le vulnerabilità del piano di controllo possano produrre effetti ben oltre il contesto locale, con ricadute tecniche, economiche e geopolitiche. Alcuni casi sono diventati emblematici: tra questi, l'hijacking del 2008 da parte di Pakistan Telecom, che rese YouTube inaccessibile a livello globale, e il caso attribuito a China Telecom, spesso richiamato per le implicazioni in termini di sicurezza e sorveglianza. L'analisi di tali incidenti consente di comprendere concretamente l'impatto delle debolezze del maggior protocollo EGP.

4.1 BGP prefix hijacking Pakistan Telecom 2008

Il 24 febbraio 2008 si verificò uno degli episodi di hijacking più noti della storia di Internet, quando AS17557 (Pakistan Telecom) annunciò in modo non autorizzato il prefisso `208.65.153.0/24`, parte dello spazio di indirizzi appartenente a YouTube (AS36561). L'azione era motivata dalla richiesta del governo pakistano di bloccare l'accesso al sito a livello nazionale, ma l'annuncio fu propagato al provider upstream AS3491 (PCCW Global), che a sua volta lo diffuse all'intera rete globale[25].

4.1.1 Timeline dell'incidente

Secondo l'analisi condotta dal RIPE NCC, l'evento ebbe inizio alle 18:47 UTC, quando il prefisso venne originato da AS17557. Nel giro di pochi minuti, gran parte degli AS mondiali preferì la nuova rotta, poiché più specifica rispetto al prefisso legittimo annunciato da YouTube (`208.65.152.0/22`). Alle 20:07 UTC YouTube tentò di contrastare l'hijack annunciando anch'essa lo stesso /24. Successivamente, dalle 20:18 UTC, pubblicò due prefissi /25 per sfruttare la regola del *longest prefix match* e riprendere il controllo del traffico. Infine, alle 21:01 UTC, PCCW ritirò gli annunci di Pakistan Telecom, riportando la situazione alla normalità[25].

4.1.2 Motivazioni dell'attacco

L'iniziativa di Pakistan Telecom non nacque come un attacco deliberato contro YouTube a livello globale, bensì come misura di censura interna. Su richiesta del governo pakistano, l'operatore tentò di rendere inaccessibile il sito all'interno del Paese, annunciando un prefisso più specifico per deviare il traffico nazionale verso un *blackhole* [1] . Tuttavia, l'annuncio non venne confinato al solo ambito domestico: propagato all'upstream provider PCCW, si diffuse rapidamente a livello internazionale. Il risultato fu che il tentativo di censura locale si trasformò in un blackout globale, con effetti non previsti e fuori dal controllo delle autorità pakistane.

4.1.3 Analisi tecnica

L'incidente fu un tipico caso di *BGP prefix hijacking*. L'origine non autorizzata di un prefisso più specifico (`/24`) rese l'annuncio di Pakistan Telecom più attraente rispetto al legittimo `/22`. Questo comportamento mise in evidenza due criticità:

- la mancanza di meccanismi di validazione sull'origine degli annunci;
- l'assenza di filtri da parte degli upstream provider, che avrebbe impedito la propagazione globale.

[1] Nel contesto del routing, un *blackhole* è una destinazione di rete fittizia in cui il traffico viene instradato e poi scartato, senza raggiungere la destinazione finale.

Il caso divenne uno studio di riferimento per l'utilizzo di strumenti di monitoraggio come **RISwhois**, **BGPlay** e **BGPath**, che consentirono di ricostruire in dettaglio la propagazione e la successiva mitigazione dell'incidente.

4.1.4 Implicazioni e lezioni apprese

L'episodio dimostrò come un'azione mirata a livello locale potesse avere effetti imprevisti e su scala globale. Per circa due ore YouTube risultò irraggiungibile in gran parte del mondo, con un impatto immediato su milioni di utenti e sull'affidabilità percepita dei servizi online. Da questo evento emerse con forza la necessità di adottare pratiche di filtraggio più rigorose, nonché lo sviluppo di soluzioni come RPKI e sistemi di rilevamento in tempo reale (es. ARTEMIS) per mitigare futuri episodi di hijacking.

4.2 BGP prefix hijacking di China Telecom 2010

Il 8 aprile 2010 si verificò un episodio che coinvolse AS4134 (China Telecom), durante il quale furono annunciati in maniera anomala circa 37.000 prefissi IP appartenenti ad altre reti sparse nel mondo. L'incidente ebbe una durata di circa 15 minuti e generò notevoli preoccupazioni a livello internazionale, non tanto per interruzioni di servizio immediate, quanto per il rischio potenziale di intercettazione e manipolazione del traffico globale[31].

4.2.1 Timeline dell'incidente

Secondo le analisi condotte dalla comunità di ricerca e riprese nel rapporto della *US–China Economic and Security Review Commission*, l'hijacking iniziò alle 15:54 UTC, quando China Telecom originò una grande quantità di prefissi non appartenenti al proprio spazio di indirizzamento. Nel giro di pochi minuti, diversi AS al di fuori della Cina instradarono parte del proprio traffico attraverso AS4134. Dopo circa un quarto d'ora gli annunci anomali cessarono, e la situazione tornò progressivamente alla normalità. Nonostante la breve durata, l'evento attirò ampia attenzione mediatica e politica per le sue implicazioni in termini di sicurezza.

4.2.2 Motivazioni e controversie

Le reali motivazioni dell'incidente non furono mai chiarite con certezza. Una parte della comunità tecnica lo interpretò come un errore di configurazione o una perdita di controllo del routing. Altri analisti ipotizzarono invece la possibilità di un'azione deliberata finalizzata a deviare traffico sensibile attraverso la Cina. Questa ambiguità alimentò un acceso dibattito internazionale: l'episodio fu citato come esempio della vulnerabilità sistemica del BGP e del rischio che un singolo operatore possa, volontariamente o meno, alterare il flusso globale dei dati.

4.2.3 Analisi tecnica

Si trattò di un classico caso di *BGP prefix hijacking*, in cui un AS annuncia prefissi non di sua proprietà. L'ampia scala dell'evento (oltre 37.000 prefissi) evidenziò come anche un'alterazione di breve durata potesse avere impatti significativi sulla topologia del routing interdominio. Dal punto di vista tecnico, l'incidente mise in luce due aspetti critici:

- la propagazione incontrollata di prefissi senza alcuna validazione di origine;
- la possibilità che il traffico venga non solo interrotto, ma anche temporaneamente instradato attraverso percorsi non previsti, aprendo scenari di potenziale intercettazione.

4.2.4 Implicazioni e lezioni apprese

L'evento non causò un'interruzione massiva di servizi, ma sollevò preoccupazioni rilevanti in termini geopolitici e di sicurezza. Il fatto che una quantità considerevole di traffico internazionale potesse transitare attraverso la Cina, anche solo per pochi minuti, fu interpretato come un campanello d'allarme sulla fragilità dell'ecosistema di routing. Il caso China Telecom 2010 contribuì a rafforzare il dibattito sulla necessità di meccanismi di validazione come RPKI e di pratiche di filtraggio più rigorose da parte degli operatori, al fine di ridurre il rischio che episodi simili possano ripetersi.

4.3 BGP route leaking di MTS Russia 2024

MTS PJSC è il principale operatore di telecomunicazioni russo, con sede a Mosca, che fornisce servizi di telefonia mobile, fissa e connettività Internet a milioni di utenti. Con oltre 80 milioni di abbonati, il suo AS (AS8359) rappresenta uno dei nodi di rete più rilevanti nell'Europa orientale. L'11 marzo 2024 MTS è stato protagonista di un significativo incidente di *BGP route leak*, che ha causato deviazioni di traffico su scala internazionale e ha evidenziato ancora una volta la fragilità strutturale del protocollo BGP di fronte a errori di configurazione.[14]

4.3.1 Timeline dell'incidente

Secondo le analisi pubblicate, il leak ebbe inizio intorno alle 07:56 UTC dell'11 marzo 2024, quando AS8359 propagò verso i propri provider e peer oltre 30.000 rotte apprese dal Hong Kong Internet Exchange (HKIX, AS4635). L'evento produsse un immediato spike di traffico sul peering BGP 8359_4635, con anomalie di reachability osservate in diverse aree della regione Asia-Pacifico, in particolare Hong Kong, Indonesia e Australia.

4.3.2 Motivazioni dell'attacco

Le cause del routing leak di MTS non sembrano riconducibili a un attacco deliberato, bensì a una configurazione impropria delle policy di instradamento. In particolare, l'anomalia potrebbe essere legata a due aspetti:

- **Filtri di export e AS-SET**: in BGP ogni operatore utilizza filtri di export per decidere quali rotte annunciare a clienti, peer e provider. Se tali filtri non sono applicati correttamente, è possibile che rotte apprese da un provider o da un peer vengano propagate a soggetti ai quali non dovrebbero essere trasmesse. Un ruolo rilevante è svolto anche dagli *AS-SET*, ovvero insiemi di Autonomous System dichiarati nei registri IRR (Internet Routing Registry). Essi servono a indicare quali rotte un AS è autorizzato a pubblicizzare. L'uso di AS-SET obsoleti, incompleti o configurati in maniera errata può portare a propagare rotte non autorizzate, favorendo fenomeni di route leaking.

- **Rapporto tra provider, peer e IXP**: per convenzione economica, un transit provider deve fornire connettività completa ai propri clienti, mentre un peer scambia solo il traffico delle proprie reti e di quelle dei clienti, senza offrire transito verso terzi. All'interno di un IXP, invece, i peer si interconnettono direttamente in un ambito limitato, scambiando esclusivamente rotte pertinenti a quel contesto. Nel caso di MTS, alcune rotte apprese in un ambiente IXP sono state propagate anche verso peer e transit provider, violando il principio del *valley-free routing* e causando la deviazione indesiderata di grandi volumi di traffico su scala internazionale.

4.3.3 Analisi tecnica

Dal punto di vista tecnico, l'evento del 11 marzo 2024 è classificabile come un tipico caso di *BGP route leaking*. MTS (AS8359) ha infatti annunciato a peer e transit provider rotte che in origine erano state apprese in un contesto differente (IXP), contravvenendo alle normali regole di propagazione.

Il traffico è stato così deviato su cammini non previsti, generando un'anomalia rilevante nella topologia del routing interdominio. L'analisi di Kentik ha mostrato come questi annunci abbiano provocato un picco di traffico mal instradato su scala globale, con un impatto significativo soprattutto verso destinazioni europee e asiatiche.

Dal punto di vista delle vulnerabilità, l'incidente ha evidenziato tre criticità principali:

- l'assenza di controlli di validazione a livello di export, che ha permesso la diffusione non autorizzata delle rotte;
- la difficoltà di monitorare in tempo reale deviazioni di traffico causate da violazioni del *valley-free routing*;
- la mancanza di meccanismi di sicurezza nativi in BGP che impediscano la propagazione di annunci impropri, come quelli tipici dei route leaks.

In sintesi, l'incidente MTS 2024 dimostra come un singolo errore di configurazione possa avere conseguenze globali, confermando la natura fragile del protocollo BGP e la necessità di adottare meccanismi di mitigazione più diffusi.

4.3.4 Implicazioni e lezioni apprese

L'incidente MTS dimostra che anche in assenza di intenzioni malevole, i *route leak* possono avere impatti significativi a livello internazionale. Le principali lezioni emerse sono:

- **Filtri di export e best practice.** È essenziale applicare controlli stringenti sulle rotte annunciate, basati su prefix lists, max-prefix limits e politiche coerenti con i ruoli customer/peer/provider.
- **Strumenti di prevenzione.** RFC 9234 introduce i *BGP Roles* e l'attributo *Only-To-Customer* (OTC), che, se adottati estensivamente, possono ridurre drasticamente la probabilità di leak accidentali.
- **Monitoraggio e risposta.** L'uso di sistemi di analisi in tempo reale (es. NetFlow, BGPStream) si è rivelato cruciale per rilevare l'anomalia e circoscriverne l'impatto.

In sintesi, l'evento ha evidenziato la vulnerabilità sistemica del piano di controllo interdominio: anche un singolo errore di configurazione, se compiuto da un operatore di grandi dimensioni, può produrre conseguenze visibili a livello globale.

4.4 Romania (DDoS mitigation provider, 2025)

Nel primo periodo di aprile 2025 un provider di mitigazione DDoS ha involontariamente diffuso annunci BGP oltre il proprio ambito previsto, causando un *route leak* che ha temporaneamente deviato una parte consistente del traffico internazionale attraverso collegamenti in Romania (Bucharest). L'incidente è stato analizzato da operatori e piattaforme di monitoraggio come Kentik e segnalato da MANRS come esempio di leak non malevolo con impatti distribuiti. [15;17]

4.4.1 Timeline dell'incidente

Secondo le analisi pubblicate, l'anomalia si è verificata nei primi giorni di aprile 2025. Kentik ha documentato come annunci originati dal provider di mitigazione siano stati riannunciati e instradati tramite Bucharest, producendo un aumento

misurabile di traffico deviato verso destinazioni in Europa e Asia. MANRS ha confermato pochi giorni dopo che non si trattava di un attacco deliberato, ma di un leak dovuto a configurazioni errate.[15;17]

4.4.2 Motivazioni dell'incidente

L'evento è stato classificato come *accidentale*. Le cause probabili sono legate a configurazioni di rete scorrette:

- **Policy di export errata**: in BGP, ogni AS stabilisce regole di export per decidere quali rotte comunicare a clienti, peer e provider. Se queste regole non sono applicate correttamente, un AS può finire per propagare rotte destinate a un ambito ristretto (ad esempio, interne a un servizio di mitigazione) a soggetti esterni, con diffusione globale.
- **Peering/IXP senza filtri**: gli Internet Exchange Point (IXP) permettono a più AS di scambiare traffico localmente. Normalmente le rotte apprese in un IXP non devono essere propagate ai transit provider, perché non sono destinate a fornire connettività universale. Nel caso di MTS, l'assenza di filtri ha consentito che rotte di questo tipo fossero diffuse al di fuori dell'ambito corretto.
- **Procedure operative inadeguate**: il provider non ha distinto con chiarezza le rotte da usare solo all'interno del servizio di mitigazione DDoS da quelle da propagare su scala globale. Questo ha favorito il leak.

4.4.3 Analisi tecnica

Tecnicamente, l'incidente è stato un *route leak*: annunci legittimi, cioè non falsificati, sono stati propagati a soggetti ai quali non erano destinati. Questo tipo di errore viola il principio del **valley-free routing**, secondo cui:

- un AS può propagare ai suoi clienti le rotte ricevute da provider e peer;
- ma non dovrebbe propagare a peer e provider le rotte ricevute da altri peer o provider.

Nel caso osservato, annunci ricevuti in un contesto locale (IXP o peering) sono stati propagati verso transit provider, creando così un percorso "non consentito" secondo il modello valley-free.

Le conseguenze pratiche osservate sono state:

- **Instradamento sub-ottimale**: parte del traffico ha seguito cammini più lunghi e meno efficienti, attraversando Bucharest anche quando non era la rotta più vicina.
- **Spike di traffico e BGP update**: i sistemi di monitoraggio hanno rilevato un aumento anomalo di traffico (NetFlow) e una grande quantità di aggiornamenti BGP, tipico segnale di un leak.
- **Congestione e instabilità**: i link coinvolti hanno rischiato di congestionarsi, mentre gli operatori hanno dovuto identificare manualmente e bloccare gli annunci non corretti.

Kentik e MANRS hanno sottolineato che questo caso mostra chiaramente come le violazioni delle regole di export possano produrre effetti globali. Tecnologie come l'attributo **OTC (Only-To-Customer)** e i **BGP Roles** introdotti da RFC 9234 permettono di specificare esplicitamente la relazione tra peer (cliente, provider, pari), riducendo così la possibilità di leak dovuti a errori umani.[15;17]

4.4.4 Implicazioni e lezioni apprese

Il caso sottolinea alcune raccomandazioni operative fondamentali:

- **Filtri di export**: applicare regole rigorose e controlli automatici per evitare che rotte locali vengano propagate globalmente.
- **Limiti di prefix (max-prefix)**: ogni sessione BGP dovrebbe avere un limite massimo di prefissi accettabili; questo serve a bloccare immediatamente eventi anomali in cui un AS annuncia troppe rotte.
- **Distinzione chiara delle rotte**: separare rotte per usi specifici (es. mitigazione DDoS) da quelle di connettività generale, in modo da non confonderle a livello di configurazione.

- **Monitoraggio in tempo reale**: usare strumenti di osservazione continua (BGP update streams, NetFlow, allarmi automatici) per individuare e reagire a un leak in pochi minuti.
- **Nuove estensioni di protocollo**: adottare meccanismi come *BGP Roles* e *OTC* (RFC 9234), che aiutano a prevenire i leak definendo ruoli chiari tra i peer BGP e limitando l'esportazione non corretta.

In conclusione, anche un errore non intenzionale da parte di un operatore legittimo può avere effetti su scala internazionale. Il caso del provider di mitigazione DDoS dell'aprile 2025 mostra come la resilienza del piano di controllo Internet dipenda non solo da strumenti di monitoraggio e intervento rapido, ma anche da regole di configurazione precise e dal supporto a nuove estensioni del protocollo. [15;17]

Capitolo 5

Prospettive future: SDN e BGP

Il BGP rimane il pilastro del routing interdominio, ma presenta limiti strutturali e di sicurezza che soluzioni come RPKI e BGPsec non hanno ancora superato, anche per la scarsa adozione su larga scala. Per questo la ricerca guarda a nuovi approcci, come la **Software-Defined Networking (SDN)**, che separa il piano di controllo da quello di inoltro e introduce flessibilità e automazione. In questo capitolo vengono presentati i principi della SDN, la sua integrazione con BGP e le possibili prospettive evolutive.[10;4;9]

5.1 Cos'è la SDN

La **Software-Defined Networking (SDN)** è un paradigma di rete che mira a superare i limiti delle architetture tradizionali, separando il *control plane* dal *data plane*. Nei router e switch convenzionali entrambe le funzioni convivono nello stesso dispositivo:

- il *control plane* elabora le decisioni di instradamento (ad es. costruendo le tabelle di routing tramite protocolli come BGP, OSPF, IS-IS);
- il *data plane* si limita a inoltrare i pacchetti secondo le regole definite.

In SDN, questa separazione viene resa esplicita: il *control plane* è centralizzato in un **controller** (software), che gestisce le decisioni di rete in maniera programmabile, mentre i dispositivi fisici svolgono solo funzioni di inoltro.[10;18]

5.1.1 Architettura e principio di separazione control/data plane

L'architettura tipica SDN prevede:

- **Application Layer**, dove applicazioni di rete (gestione QoS, sicurezza, monitoring) interagiscono col controller;
- **Control Layer**, rappresentato dal controller SDN, che calcola le politiche e traduce le richieste in regole concrete;
- **Infrastructure Layer**, composto da switch e router che eseguono le regole.

Il principio guida è la programmabilità: le decisioni di instradamento non sono più vincolate al comportamento distribuito dei singoli dispositivi, ma possono essere determinate in modo globale.

5.1.2 Vantaggi principali: flessibilità, programmazione, automazione

I vantaggi riconosciuti dalla comunità scientifica e dagli operatori sono molteplici:

- **Flessibilità**: possibilità di aggiornare le politiche di rete in tempo reale senza riconfigurare manualmente ogni dispositivo;
- **Programmazione**: la rete diventa programmabile via API, facilitando l'integrazione con servizi e applicazioni;
- **Automazione**: compiti complessi (bilanciamento del carico, rerouting in caso di guasti) possono essere automatizzati, riducendo errori umani e tempi di reazione.

Queste caratteristiche rendono SDN una piattaforma ideale per affrontare alcune delle sfide di scalabilità e sicurezza non risolte dal BGP.[10]

5.2 Integrazione tra SDN e BGP

5.2.1 Routing interdominio gestito centralmente

Il BGP, progettato come protocollo distribuito e basato sulla fiducia tra operatori, presenta limiti strutturali: non offre validazione nativa degli annunci e le soluzioni proposte (come RPKI e BGPsec) non sono ancora adottate su larga scala. Le ragioni di questa adozione limitata includono:

- costi di implementazione e complessità gestionale per gli operatori;
- compatibilità con l'infrastruttura esistente;
- preoccupazioni per la centralizzazione del trust (ad es. nel modello RPKI).

In questo contesto, l'SDN offre una prospettiva alternativa: il controller potrebbe monitorare in tempo reale le sessioni BGP, validare le rotte e imporre politiche di instradamento uniformi a livello interdominio. [9]

5.2.2 Esempi di progetti o framework (es. SDX, BGP-SDN)

Alcuni progetti hanno esplorato l'integrazione tra SDN e BGP:

- **SDX (Software Defined Internet Exchange)**: sperimentazioni presso IXPs hanno mostrato come i controller SDN possano offrire funzionalità avanzate di peering e policy enforcement, andando oltre le capacità del BGP tradizionale. [4]
- **BGP-SDN**: approcci ibridi che consentono al controller SDN di interagire con i router BGP, fornendo un livello aggiuntivo di controllo centralizzato senza stravolgere l'architettura esistente. Questi framework mirano a introdurre sicurezza e automazione progressivamente, riducendo i rischi di incompatibilità.

5.3 Prospettive evolutive

5.3.1 Reti programmabili e scenari futuri

La convergenza tra SDN e BGP si inserisce nel più ampio scenario delle **reti programmabili**. La visione futura è quella di un ecosistema Internet in cui:

- le decisioni di instradamento possano essere influenzate dinamicamente da parametri di performance, sicurezza o policy economiche;
- i controlli di sicurezza (es. validazione degli annunci, rilevamento anomalie) vengano implementati direttamente nei controller SDN;
- l'interoperabilità con protocolli legacy come BGP resti garantita, ma con maggiore capacità di supervisione.

Questo approccio promette di superare i limiti intrinseci del BGP distribuito, ma solleva interrogativi su resilienza, scalabilità e governance globale.

5.3.2 Possibili impatti su sicurezza e gestione globale

Dal punto di vista della sicurezza, l'uso di SDN può consentire:

- validazione più rapida di annunci sospetti;
- isolamento immediato di rotte malevole o errate;
- coordinamento centralizzato nella risposta a incidenti di hijacking o leak.

D'altra parte, la centralizzazione comporta rischi: un controller compromesso o malfunzionante potrebbe avere effetti devastanti. Inoltre, la governance di soluzioni centralizzate a livello interdominio solleva questioni di fiducia simili a quelle di RPKI.

Sintesi critica. La letteratura converge su un punto: le vulnerabilità del BGP rappresentano una sfida ancora aperta. RPKI e BGPsec sono strumenti importanti, ma la loro adozione rimane lenta e incompleta. Le ragioni sono molteplici:

- **Costi operativi e complessità**: l'implementazione richiede aggiornamenti software, gestione di certificati crittografici e modifiche nei processi interni degli operatori, scoraggiando una diffusione rapida.

- **Problemi di interoperabilità**: non tutti i router e sistemi legacy supportano pienamente RPKI o BGPsec, rendendo difficile l'adozione senza sostituire infrastrutture critiche.
- **Centralizzazione del trust**: in RPKI l'autorità di certificazione è gerarchica e ciò genera timori di concentrazione del potere e rischi legati a errori o compromissioni dei registri.
- **Impatto sulle performance**: BGPsec, basato su firme crittografiche per ogni hop, introduce un overhead computazionale significativo che solleva dubbi sulla scalabilità globale.

Le soluzioni SDN offrono prospettive di maggiore controllo e programmabilità, ma sollevano a loro volta interrogativi di sicurezza e governance. Le direzioni più promettenti sembrano essere quelle ibride: mantenere la compatibilità con l'infrastruttura esistente, introducendo progressivamente funzionalità centralizzate (validazione, enforcement delle policy) e promuovendo standard interoperabili che bilancino sicurezza, resilienza e decentralizzazione.

Appendice A

Ricerca su ECDSA

Capitolo 2

Firma digitale con curve ellittiche: ECDSA

2.1 Nascita di ECDSA

L'ECDSA nasce come estensione dell'algoritmo Digital Signature Algorithm (DSA), sviluppato agli inizi degli anni '90 dal National Institute of Standards and Technology (NIST) degli Stati Uniti come standard federale per la firma digitale (Federal Information Processing Standard (FIPS) 186 nel 1991).

Successivamente, grazie agli studi di due matematici americani, Neal Koblitz e Victor Miller, fu dimostrato che le curve ellittiche potevano essere sfruttate per costruire sistemi crittografici a chiave pubblica con una sicurezza equivalente a quella di algoritmi preesistenti (come RSA o lo stesso DSA), ma con chiavi significativamente più piccole e, di conseguenza, operazioni computazionali più rapide ed efficienti.

Fu Scott Vanstone, matematico canadese e co-fondatore dell'azienda Certicom, a proporre specificamente l'uso delle curve ellittiche all'interno dello schema DSA. La sua proposta, formulata nei primi anni '90, portò alla definizione dell'ECDSA, che fu infine standardizzato nel 1999 dallo Institute of Electrical and Electronics Engineers (IEEE) (nello standard P1363) e dall'American National Standards Institute (ANSI) (nello standard X9.62), segnando l'ingresso ufficiale di ECDSA nel panorama della crittografia moderna.

2.2 Firma digitale

ECDSA è un tipo particolare di firma digitale che utilizza le curve ellittiche. Quindi, prima di addentrarci nel comprendere il suo funzionamento, è necessario parlare di firme digitali e delle hash function (utilizzate per la creazione delle firme).

2.2.1 Hash Function

Le hash function sono particolari funzioni che prendono un input di lunghezza variabile (l'input può avere anche più bit di quelli fissi dell'output), e restituisce un output di lunghezza fissa in base al tipo di funzione di hash. L'output di una hash function prende il nome di **message digest**, **digest** o **impronta**.

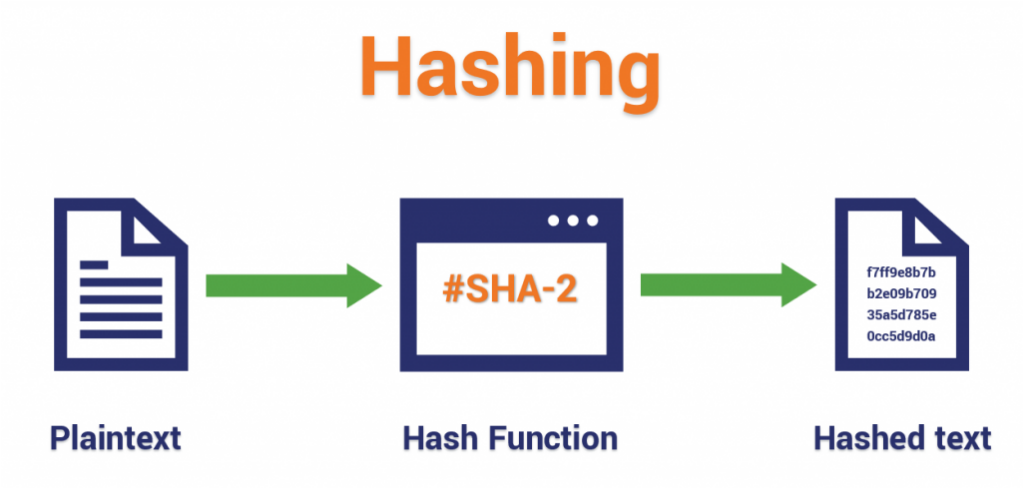

Figura 2.1: Hash function

Caratteristiche hash function

- è **one way function** → dato l'output, non è possibile risalire all'input
- è **deterministica** → dato lo stesso input, per la medesima funzione, ottengo sempre lo stesso output
- gode del cosiddetto "**effetto valanga**" → se in input cambia anche solo di 1 bit, cambia completamente l'output.

2.2.2 Come avviene la firma digitale e cosa garantisce

Mittente

1. Si parte dal **messaggio in chiaro** (plaintext), indicato con M.
2. Il mittente calcola il **digest** del messaggio tramite una funzione di hash sicura: $h = \text{Hash}(M)$.
3. Il digest h viene **firmato con la chiave privata del mittente** SK_A (private key), ottenendo la **firma digitale**: $\sigma = \text{Sign}_{SK_A}(h)$.
4. Il mittente invia al destinatario la coppia: (M, σ).

Destinatario

1. Il destinatario riceve (M, σ) e calcola nuovamente il digest: $h' = \text{Hash}(M)$.
2. Il destinatario verifica la firma usando la **chiave pubblica del mittente** PK_A: $h = \text{Verify}_{PK_A}(\sigma)$.
3. La firma è valida se e solo se: $h = h'$.

Se $h = h'$ allora è garantita l'integrità del messaggio (grazie all'hash function) e l'autenticazione del mittente, oltre che il non ripudio (ovvero il mittente non può negare di aver firmato il messaggio, dato che solo lui possiede la chiave privata).

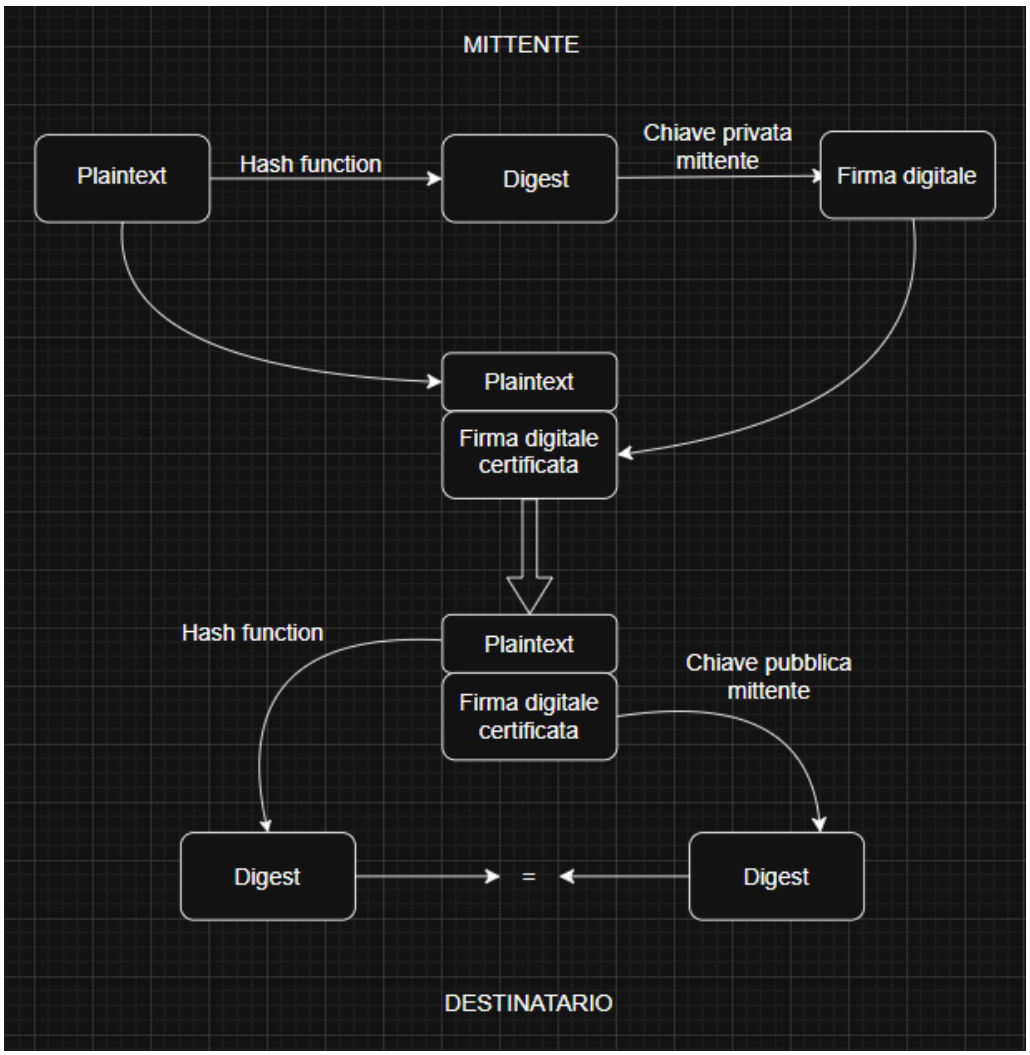

Figura 2.2: Firma digitale

2.3 Come funziona ECDSA

La firma digitale tramite ECDSA si articola in 3 fasi che andremo ad analizzare una per una: Inizializzazione dei parametri, generazione della firma da parte del mittente e verifica della firma da parte del destinatario.

2.3.1 Inizializzazione dei parametri

Questi parametri sono pubblici e devono essere concordati da tutte le parti coinvolte nella comunicazione. Non vengono scelti da ogni singolo utente ma sono scelti da standard predefiniti come quelli NIST (es. curve P-192, P-256, P-384, P-521) o Standard for Efficient Cryptography Group (SECG) (es. secp256k1, usata in Bitcoin). Le curve sono definite su campi finiti di grandi dimensioni, come $\mathbb{F}_p$ con p primo da 256 o più bit.

- Un campo finito primo $\mathbb{F}_p$, con $p > 3$ primo.
- Una curva ellittica E definita su $\mathbb{F}_p$, espressa in forma ridotta di Weierstrass:

$$E : y^2 = x^3 + ax + b \mod p$$

con $a, b \in \mathbb{F}_p$, tali che il discriminante $\Delta = 4a^3 + 27b^2 \mod p \neq 0$, per garantire la non-singolarità.

- Un punto generatore $G \in E(\mathbb{F}_p)$ di ordine primo n, tale che:

$$nG = \mathcal{O}$$

dove $\mathcal{O}$ è il punto all'infinito.

- Una funzione di hash crittografica $H : \{0,1\}^* \to \{0,1\}^l$, come SHA-256. L'output dell'hash viene utilizzato per derivare un valore numerico e (spesso z o H_M) che è poi utilizzato nel calcolo della firma. Per l'ECDSA, la lunghezza dell'output dell'hash deve essere almeno pari alla dimensione in bit di n. Tipicamente, il valore hash è convertito in un intero e poi, se necessario, troncato o ridotto modulo n.

Chiavi

Ogni utente genera le proprie chiavi basandosi sui parametri di dominio pubblico stabiliti:

- **Chiave Privata (d):** Un intero d scelto casualmente e in modo sicuro dall'intervallo $d \in_R \{1, \ldots, n-1\}$. Questa chiave deve rimanere segreta.
- **Chiave Pubblica (Q):** Il punto $Q = dG \in E(\mathbb{F}_p)$, calcolato moltiplicando la chiave privata d per il punto generatore G. La chiave pubblica Q può essere distribuita liberamente.

2.3.2 Generazione della firma

Supponiamo che l'utente A (mittente) voglia firmare un messaggio m utilizzando la sua chiave privata d. I passaggi della generazione della firma, secondo lo standard ECDSA, sono i seguenti:

1. **Calcolo del Digest del Messaggio:** Si calcola il digest crittografico del messaggio m usando la funzione di hash H:

$$e = H(m)$$

Questo valore e viene poi interpretato come un intero. Il valore z utilizzato nel calcolo della firma è derivato da e. Se la lunghezza in bit di e supera la lunghezza in bit dell'ordine n, si utilizzano i bit più significativi di e per formare z, in modo che la sua lunghezza in bit sia pari a quella di n (ovvero z è l'intero rappresentato dai $\lfloor \log_2 n \rfloor$ bit più significativi di e, o semplicemente e se $e < n$).

2. **Scelta del Nonce [1] Crittografico:** Si sceglie un numero intero casuale e segreto $k \in_R \{1, \ldots, n-1\}$. È **fondamentale** che k sia generato in modo crittograficamente sicuro per ogni singola firma e non venga mai, per nessuna ragione, riutilizzato per firme diverse. La compromissione o la riutilizzazione di k esporrebbe immediatamente la chiave privata d.

[1] Il termine "nonce" sta per "number used once".

3. **Calcolo del Punto sulla Curva:** Si calcola il punto $(x_1, y_1) = [k]G$ eseguendo una *moltiplicazione scalare* del punto generatore G per l'intero k.

4. **Calcolo di r:** Si calcola la prima componente della firma, r, interpretando la coordinata x_1 come un intero:

$$r = x_1 \mod n$$

   Se $r = 0$, si deve scartare il valore k corrente e ripetere il processo a partire dal passaggio 2 con un nuovo k.

5. **Calcolo di s:** Si calcola l'inverso moltiplicativo di k modulo n, denotato k^{-1} (cioè $k \cdot k^{-1} \equiv 1 \pmod{n}$). Successivamente, si calcola la seconda componente della firma, s:

$$s = k^{-1}(z + dr) \mod n$$

   Se $s = 0$, anche in questo caso si scarta il valore k corrente e si ripete il processo a partire dal passaggio 2 con un nuovo k.

Output: La firma è la coppia ordinata (r, s). Entrambi i valori sono elementi di $\mathbb{Z}_n^*$ (il gruppo moltiplicativo degli interi modulo n) e insieme rappresentano la firma digitale di m.

Al destinatario vengono spediti sia m che la corrispondente firma digitale (r, s).

2.3.3 Verifica della firma

Supponiamo che il destinatario riceva un messaggio m in chiaro insieme alla firma digitale (r, s). Per verificare l'autenticità e l'integrità del messaggio (e che sia stato effettivamente firmato dal mittente), il destinatario esegue i seguenti passaggi. Questo processo richiede la conoscenza della **chiave pubblica del mittente** Q.

1. **Controllo dei Valori della Firma:** Si verifica innanzitutto che le componenti della firma r e s siano entrambe valide, ovvero che appartengano all'intervallo $[1, n-1]$. Se $r < 1$ o $r \geq n$, oppure se $s < 1$ o $s \geq n$, la firma è considerata **invalida** e il processo di verifica si interrompe.

2. **Calcolo del Digest del Messaggio:** Il destinatario calcola il digest crittografico del messaggio ricevuto m utilizzando la stessa funzione di hash H impiegata dal mittente:

$$e = H(m)$$

   Come nel processo di generazione della firma, questo valore e viene interpretato come un intero. Il valore z utilizzato nei calcoli successivi è derivato da e. Se la lunghezza in bit di e supera la lunghezza in bit dell'ordine n, si utilizzano i bit più significativi di e per formare z, in modo che la sua lunghezza in bit sia pari a quella di n.

3. **Calcolo dell'Inverso Modulare di s:** Si calcola l'inverso moltiplicativo di s modulo n, denotato w. Questo valore soddisfa la relazione $s \cdot w \equiv 1 \pmod{n}$. Tale calcolo viene tipicamente eseguito utilizzando l'algoritmo di Euclide Esteso.

$$w = s^{-1} \mod n$$

4. **Calcolo dei Coefficienti Intermedi:** Si calcolano due coefficienti ausiliari, u_1 e u_2, che saranno usati nel prossimo passaggio per ricostruire un punto sulla curva:
$$u_1 = z \cdot w \mod n$$
$$u_2 = r \cdot w \mod n$$

5. **Calcolo del Punto di Verifica sulla Curva:** Utilizzando i coefficienti u_1 e u_2, il punto generatore G, e la chiave pubblica del mittente Q, si calcola un punto (x_V, y_V) sulla curva. Questa operazione coinvolge due **moltiplicazioni scalari di punti** ($[u_1]G$ e $[u_2]Q$) e una successiva **addizione di punti** sulla curva ellittica:
$$(x_V, y_V) = [u_1]G + [u_2]Q$$
Se il risultato di questa operazione è il punto all'infinito $\mathcal{O}$, la firma è considerata invalida. [2]

6. **Verifica Finale:** Infine, si confronta la coordinata x_V del punto calcolato nel passaggio precedente con il valore r fornito nella firma. La verifica ha successo se e solo se la coordinata x_V, interpretata come un intero e ridotta modulo n, è uguale a r:
$$r \equiv x_V \mod n$$

Conclusione: Se l'uguaglianza nel passaggio 6 è verificata, la firma è considerata **valida**. Questo successo implica che:

- Il messaggio non è stato modificato dopo la firma (garanzia di **integrità**).
- Il messaggio è stato firmato con la chiave privata corrispondente alla chiave pubblica Q fornita (garanzia di **autenticità** del mittente).
- Il mittente non può negare di aver firmato il messaggio, poiché solo lui possedeva la chiave privata d necessaria a generare r e s in modo congruente con Q (garanzia di **non ripudio**).

2.3.4 Esempio generazione e verifica firma

Per comprendere meglio il funzionamento dell'algoritmo, vediamo un esempio semplificato con numeri piccoli [3].

Parametri pubblici condivisi

- Campo finito: $\mathbb{F}_{17}$
- Curva ellittica: $E : y^2 = x^3 + 2x + 2 \mod 17$
- Punto generatore: $G = (5, 1)$, con ordine primo $n = 19$
- Funzione di hash: $H(m) = 9$ (supponiamo che $H(\text{“ciao”}) = 9$)

[2] Questo caso è estremamente raro se k è scelto correttamente

[3] Questi numeri non garantiscono alcuna sicurezza crittografica ma servono a capire meglio il funzionamento dell'algoritmo.

Chiavi

- Chiave privata $d = 7$
- Chiave pubblica $Q = dG = 7G$ (supponiamo $Q = (6, 3)$)

Generazione della firma

1. Calcolo dell'hash del messaggio:
$$e = H(m) = 9 \Rightarrow z = 9$$
2. Scelta del nonce segreto:
$$k = 3$$
3. Calcolo del punto kG:
$$kG = 3G = (10, 6) \Rightarrow x_1 = 10$$
4. Calcolo di r:
$$r = x_1 \mod n = 10 \mod 19 = 10$$
5. Calcolo di s:
Calcoliamo l'inverso di $k \mod n$: $k^{-1} = 13 \mod 19$, poiché $3 \cdot 13 = 39 \equiv 1 \mod 19$.

$$s = k^{-1}(z+dr) \mod n = 13(9+7\cdot 10) \mod 19 = 13\cdot 79 \mod 19 = 1027 \mod 19 = 1$$

Firma generata: $(r, s) = (10, 1)$

Verifica della firma

1. Verifica che $r, s \in [1, n-1]$: OK
2. Calcolo dell'hash:
$$z = H(m) = 9$$
3. Calcolo dell'inverso di s:
$$w = s^{-1} \mod n = 1^{-1} \mod 19 = 1$$
4. Calcolo dei coefficienti:
$$u_1 = z \cdot w \mod n = 9 \cdot 1 \mod 19 = 9$$
$$u_2 = r \cdot w \mod n = 10 \cdot 1 \mod 19 = 10$$
5. Calcolo del punto:
$$(x_V, y_V) = [u_1]G + [u_2]Q = 9G + 10Q$$
Supponiamo che $9G = (3, 1)$, $10Q = (7, 14)$ e che la somma valga $(10, 6)$
6. Verifica finale:
$$x_V \mod n = 10 \mod 19 = 10 = r \Rightarrow \text{firma valida}$$

Conclusione: La firma $(10, 1)$ è corretta e verificabile, confermando l'integrità e l'autenticità del messaggio firmato.

2.4 Analisi della sicurezza di ECDSA

In questa sezione, esploreremo l'importanza della casualità di k, vedremo alcuni attacchi noti a ECDSA e le best practices per evitarli. Infine, andremo ad analizzare come ECDSA resiste o meno in uno scenario con attaccanti dotati di computer ed algoritmi quantistici.

2.4.1 Importanza della casualità di k

Nel processo di firma ECDSA, il valore k è un numero intero scelto in modo casuale per ogni firma. La sicurezza dell'intero schema dipende fortemente dalla **non ripetibilità** e **non prevedibilità** di questo valore. La sua gestione scorretta compromette direttamente la **riservatezza della chiave privata** d.

Riutilizzo di k

Se lo stesso valore k viene usato per firmare due messaggi distinti m_1 e m_2, generando due firme (r, s_1) e (r, s_2), allora un attaccante può calcolare k nel seguente modo:

$$k = \frac{z_1 - z_2}{s_1 - s_2} \mod n$$

dove $z_1 = H(m_1)$, $z_2 = H(m_2)$, e l'inverso è calcolato modulo n. Una volta noto k, si può derivare la chiave privata d da una delle firme:

$$d = \frac{s \cdot k - z}{r} \mod n$$

Esempio pratico: l'attacco alla PlayStation 3 (2010)

Un attacco emblematico alla sicurezza di ECDSA si è verificato nel 2010 con la compromissione della *Sony PlayStation 3*. In quell'occasione, i ricercatori scoprirono che il firmware della console impiegava un generatore di numeri pseudo-casuali mal progettato per calcolare il nonce k usato nelle firme ECDSA. Invece di generare un valore di k completamente casuale e imprevedibile, come richiesto per la sicurezza dell'algoritmo, il sistema produceva valori *identici o altamente prevedibili* ogni volta che veniva generata una firma.

Ciò ha portato a una rottura completa del sistema di sicurezza della console: gli hacker (ovvero il gruppo fail0verflow e George Hotz) sono stati in grado di firmare codice come se fossero Sony stessa, eludendo le protezioni contro software non autorizzato.

Predizione parziale di k

Anche la **conoscenza parziale dei bit di** k—ad esempio ottenuta tramite attacchi side-channel—può essere sufficiente per risalire a d, utilizzando tecniche di riduzione reticolare (es. attacchi Lattice basati su LLL [4]). Ciò accade perché la relazione che lega s, k, r, d e z può essere trasformata in un'istanza di un problema di approssimazione di vettori corti (SVP), risolvibile se abbastanza bit di k sono noti.

[4] L'algoritmo LLL (Lenstra–Lenstra–Lovász) permette di risolvere problemi di reticoli ed è impiegato in crittoanalisi con conoscenza parziale di bit.

Contromisure

Per mitigare questi attacchi è fondamentale che k sia:

- generato con una **sorgente di entropia crittograficamente sicura**;
- **unico** per ogni messaggio firmato;
- **imprevedibile**, anche parzialmente.

Lo standard RFC 6979 propone una variante detta *Deterministic ECDSA*, in cui k è generato **in modo deterministico** a partire dal messaggio m (più precisamente dal suo hash $z = H(m)$) e dalla chiave privata d, tramite una funzione pseudo-casuale crittograficamente sicura come HMAC-DRBG. Questo significa che, per lo stesso messaggio e la stessa chiave privata, il valore di k sarà sempre lo stesso, ma **inaccessibile a un attaccante** che non conosce d. In questo modo si elimina completamente la dipendenza da generatori casuali esterni, riducendo drasticamente il rischio di errori implementativi.

2.4.2 Attacchi noti a ECDSA

La robustezza teorica di ECDSA non protegge automaticamente da tutte le minacce: molte vulnerabilità risiedono a livello implementativo, dove un attaccante può sfruttare informazioni ausiliarie o manipolazioni hardware per compromettere la segretezza delle chiavi. In questa sezione vengono illustrati alcuni attacchi pratici noti.

Side-Channel Attacks

Questi attacchi si basano sull'analisi delle informazioni fisiche emesse durante l'esecuzione dell'algoritmo. Non mirano alla rottura matematica di ECDSA, ma alla deduzione di bit sensibili osservando fenomeni collaterali:

- **Timing attacks:** nelle implementazioni di ECDSA, la computazione che coinvolge le operazioni matematiche possono richiedere tempi leggermenti diversi a seconda dei bit 0 o 1 del nonce k. Se l'implementazione non è a tempo costante, ovvero se non impiega sempre lo stesso tempo indipendentemente dai dati, l'attaccante può misurare il tempo che il dispositivo impiega per firmare. Basandosi su queste minuscole differenze di tempo, può così dedurre quali bit erano 0 e quali 1, ricostruendo una parte o l'intero valore di k.
- **Simple Power Analysis (SPA):** questa tecnica sfrutta la correlazione diretta tra le operazioni e il loro consumo di energia. Durante la generazione della firma ECDSA, una delle operazioni fondamentali è il calcolo del punto $R = [k]G$, ottenuto tramite una moltiplicazione scalare del punto generatore G per l'intero segreto k. Questa operazione viene tipicamente realizzata con l'algoritmo *double-and-add* (vedi sezione 1.2.2). All'interno dell'algoritmo, le operazioni di *doubling* e *addition* hanno consumi energetici distinti, osservabili con un oscilloscopio o un'analisi delle tracce di potenza. Un attaccante può quindi ricostruire la sequenza dei bit di k osservando il pattern di consumo energetico del dispositivo durante l'esecuzione dell'algoritmo. Se k viene compromesso in questo modo, anche la chiave privata d può essere ricostruita.

- **Differential Power Analysis (DPA):** la DPA non si di basa sull'osservazione di una singola operazione o di un singolo pattern evidente, ma sfrutta le *piccole differenze statistiche* nel consumo energetico che emergono quando il dispositivo elabora dati diversi.

  Il processo tipico della DPA prevede i seguenti passaggi:

  - **Raccolta di molte tracce di potenza:** Vengono registrate centinaia o migliaia di tracce di consumo energetico mentre il dispositivo esegue la stessa operazione (es. una cifratura), ma con *dati di input diversi* (e spesso anche la stessa chiave segreta).
  - **Predizione del consumo:** L'attaccante formula delle ipotesi sui valori intermedi che un dispositivo dovrebbe assumere durante un'operazione specifica, basandosi su una *chiave parziale ipotizzata*. Ad esempio, in una cifratura AES, si può ipotizzare una parte della chiave e calcolare il valore del bit più significativo dell'output di uno S-box per ogni traccia.
  - **Divisione delle tracce:** Le tracce di potenza raccolte vengono divise in due gruppi distinti, basandosi sul valore ipotizzato del bit intermedio (es. gruppo A se il bit è 0, gruppo B se il bit è 1).
  - **Calcolo della differenza:** Viene calcolata la differenza media tra i profili energetici dei due gruppi (A e B) per ogni punto temporale.
  - **Identificazione della chiave corretta:** Se l'ipotesi sulla chiave parziale è corretta, la differenza media mostrerà un "picco" significativo in corrispondenza del momento in cui l'operazione che dipende da quel bit viene eseguita. Se l'ipotesi è sbagliata, la differenza sarà prossima allo zero. Ripetendo questo processo per tutte le possibili chiavi parziali, l'attaccante può identificare la chiave segreta.

2.4.3 Best practices per un uso sicuro di ECDSA

Per garantire un uso sicuro di ECDSA, è fondamentale adottare alcune best practices tra cui la generazione sicura e non riutilizzabile di k (es. RFC 6979), l'uso di curve standard e la verifica dei punti sulla curva. Inoltre, vediamo nel dettaglio come prevenire i vari side-attacks descritti nella sezione superiore.

- **Time attacks:** per prevenire questo tipi di attacchi, la cosa migliore è utilizzare implementazioni a tempo costante di tutte le operazioni, indipendentemente dai bit di k.

- **SPA:** le contromisure a questo attacco prevedono di rendere indistinguibili le operazioni o a mascherare le informazioni reali su k.

  - **Offuscamento algoritmico:** anziché calcolare $[k]G$ direttamente, si calcola $[k + r \cdot n]G$ dove k è lo scalare segreto, r è un numero intero casuale generato al momemento e n è l'ordine della curva ellittica (è pubblico). In questo modo, dato che $nG = O$ (il punto all'infinito), allora $r \cdot nG = O$. Di conseguenza, $[k + r \cdot n]G = [k]G + [r \cdot n]G = [k]G + O = [k]G$. Il risultato finale quindi non cambia, ma cambia la sequenza di operazioni eseguite, che in questo modo dipende da r che varia ad ogni esecuizione, rendendo SPA inutilizzabile.

- **Utilizzo di algoritmi uniformi:** un'alternativa all'offuscamento algoritmico, è l'utilizzo di alcuni algoritmi che sono progettati per eseguire una sequenza di operazioni che è indipendente dal valore dei bit di k.
  - **Scala di Montgomery:** durante il calcolo della moltiplicazione scalare $[k]G$, l'algoritmo lavora con due punti diversi in parallelo invece di uno solo, tipicamente P_0 e P_1. Per ogni bit di k, vengono sempre eseguite sia un'operazione di doubling che un'operazione di addition, indipendentemente dal valore del bit. La selezione dei punti su cui operare e la riassegnazione dei risultati avvengono tramite meccanismi a tempo costante (es. swap condizionali), eliminando le ramificazioni dipendenti dal dato segreto. Questo impedisce all'attaccante di distinguere i bit di k basandosi su variazioni nella traccia di potenza.
  - **Double-and-Add Always:** questo algoritmo prevede l'esecuzione di entrambe le operazioni di *doubling* e *addition* dell'algoritmo *double-and-add* per ogni bit di k, sia che il bit sia 0 che 1. In questo modo viene eliminata la dipendenza della traccia energetica dal valore del bit. (C'è del lavoro inutile alla computazione dell'algoritmo).

- **DPA:** le contromisure a questo tipo di attacco mirano a rompere la correlazione tra i dati intermedi (che dipendono dalla chiave segreta) e il consumo energetico osservabile. Le tecniche principali includono:
  - **Randomizzazione dei dati intermedi:** consiste nell'introdurre variabili casuali nei calcoli interni (es. maschere additive o moltiplicative su k e sui punti della curva), in modo che i valori intermedi cambino a ogni firma, pur mantenendo invariato il risultato finale.
  - **Coordinate randomizzate:** invece di usare le coordinate affini (x, y), si possono usare coordinate proiettive o Jacobiane con randomizzazione. Ad esempio, si moltiplica il punto iniziale G per uno scalare casuale prima della moltiplicazione scalare.
  - **Point blinding:** si calcola $[k]G$ come $[k](G + [r]G) - [k][r]G$, dove r è un intero casuale scelto ogni volta. Questa tecnica "nasconde" le operazioni reali rendendo più difficile isolare il contributo diretto di k.
  - **Scalar blinding:** anziché usare direttamente k, si usa $k' = k + rn$, dove r è casuale e n è l'ordine del punto G. Poiché $nG = \mathcal{O}$, il risultato rimane $[k']G = [k]G$, ma la sequenza di operazioni cambia, rendendo inefficace l'analisi statistica.

Riferimenti bibliografici

[1] Cloudflare, Inc. What is an autonomous system?, n.d. Accessed: 2025-07-05. URL: https://www.cloudflare.com/learning/network-layer/what-is-an-autonomous-system/.

[2] DataPath.io. The history of border gateway protocol, 2016. Accessed: 2025-07-26. URL: https://medium.com/@datapath_io/the-history-of-border-gateway-protocol-a212b7ee6208.

[3] Fortinet. Tcp/ip model vs. osi model, 2024. URL: https://www.fortinet.com/resources/cyberglossary/tcp-ip-model-vs-osi-model.

[4] Arpit Gupta, Laurent Vanbever, Muhammad Shahbaz, Sean P. Donovan, Nick Feamster, Jennifer Rexford, and Scott Shenker. Sdx: A software defined internet exchange. In *Proceedings of the ACM SIGCOMM 2014 Conference*, pages 551–562, 2014. doi:10.1145/2619239.2626300.

[5] John Hawkinson and Tony Bates. Guidelines for creation, selection, and registration of an autonomous system, 1996. RFC 1930, IETF. URL: https://datatracker.ietf.org/doc/html/rfc1930.

[6] Nils Höger, Nils Rodday, and Gabi Dreo Rodosek. Mitigating bgp route leaks with attributes and communities: A stopgap solution for path plausibility, 2025. Accessed: 2025-08-23. URL: https://onlinelibrary.wiley.com/doi/10.1002/nem.70002, doi:10.1002/nem.70002.

[7] Internet Assigned Numbers Authority. IANA number resources, 2024. Accessed July 5, 2025. URL: https://www.iana.org/numbers.

[8] S. Kent, C. Lynn, and K. Seo. An Infrastructure to Support Secure Internet Routing. RFC 6480, 2012. Accessed: 2025-08-26. URL: https://www.rfc-editor.org/rfc/rfc6480, doi:10.17487/RFC6480.

[9] Hyojoon Kim, Jennifer Rexford, et al. Efficient inter-domain routing with sdn and bgp integration. In *Proceedings of IEEE HotSDN Workshop*, 2017. Example reference on SDN-BGP integration.

[10] Diego Kreutz, Fernando Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, 2015. doi:10.1109/JPROC.2014.2371999.

[11] M. Lepinski and K. Sriram. BGPsec Protocol Specification. RFC 8205, 2017. Accessed: 2025-08-24. URL: https://www.rfc-editor.org/rfc/rfc8205, doi:10.17487/RFC8205.

[12] Kirk Lougheed and Yakov Rekhter. RFC 1163: A Border Gateway Protocol (BGP). https://datatracker.ietf.org/doc/html/rfc1163, 1990. Accessed: 2025-07-26.

[13] Kirk Lougheed and Yakov Rekhter. RFC 1267: A Border Gateway Protocol 3 (BGP-3). https://datatracker.ietf.org/doc/html/rfc1267, 1991. Accessed: 2025-07-26.

[14] Doug Madory. Bgp routing leak leads to spike of misdirected traffic, March 2024. Kentik Blog, accesso: 16 settembre 2025. URL: https://www.kentik.com/analysis/BGP-Routing-Leak-Leads-to-Spike-of-Misdirected-Traffic.

[15] Doug Madory. Beyond their intended scope: Ddos mitigation leak, April 2025. Kentik blog; Accessed: 16 September 2025. URL: https://www.kentik.com/blog/beyond-their-intended-scope-ddos-mitigation-leak/.

[16] ManageEngine. Border gateway protocol (bgp), 2024. URL: https://www.manageengine.com/it-operations-management/border-gateway-protocol.html.

[17] MANRS. Beyond their intended scope: Ddos mitigation leak (commentary), April 2025. MANRS blog; Accessed: 16 September 2025. URL: https://manrs.org/2025/04/ddos-mitigation-leak-kentik/.

[18] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 69–74, 2008. doi:10.1145/1355734.1355746.

[19] J. Mitchell and J. G. Scudder. Autonomous system (as) reservation for private use, 2011. RFC 6996, IETF. URL: https://datatracker.ietf.org/doc/html/rfc6996.

[20] Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, and Alberto Dainotti. Bgpstream: A software framework for live and historical bgp data analysis. In *Proceedings of the 2016 Internet Measurement Conference (IMC)*, pages 437–444. ACM, 2016. URL: https://dl.acm.org/doi/10.1145/2987443.2987482, doi:10.1145/2987443.2987482.

[21] Samuel Brako Oti and Joseph Hayfron-Acquah. Practical security approaches against border gateway protocol (bgp) session hijacking attacks between autonomous systems. *Communications and Network*, 6(3):167–176, 2014. URL: https://www.scirp.org/journal/paperinformation.aspx?paperid=46857, doi:10.4236/cn.2014.63019.

[22] Yakov Rekhter and Tony Li. RFC 1654: A Border Gateway Protocol 4 (BGP-4). https://datatracker.ietf.org/doc/html/rfc1654, 1994. Accessed: 2025-07-26.

[23] Yakov Rekhter, Tony Li, and Susan Hares. RFC 4271: A Border Gateway Protocol 4 (BGP-4). https://datatracker.ietf.org/doc/html/rfc4271, 2006. Accessed: 2025-07-26.

[24] Yakov Rekhter and Kirk Lougheed. RFC 1105: Border Gateway Protocol (BGP). https://datatracker.ietf.org/doc/html/rfc1105, 1989. Accessed: 2025-07-26.

[25] RIPE NCC. Youtube hijacking: A ripe ncc ris case study. Technical report, RIPE Network Coordination Centre, March 2008. Accessed: 2025-08-27. URL: https://www.ripe.net/about-us/news/youtube-hijacking-a-ripe-ncc-ris-case-study.

[26] RIPE NCC. Routing information service (ris), 2025. Accessed: 2025-08-26. URL: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris.

[27] Eric C. Rosen. RFC 904: Exterior Gateway Protocol (EGP). https://datatracker.ietf.org/doc/html/rfc904, 1984. Accessed: 2025-07-26.

[28] Kotikalapudi Sriram, Doug Montgomery, Danny McPherson, Eric Osterweil, and Marla Azinger. RFC 7908: Problem Definition and Classification of BGP Route Leaks. https://datatracker.ietf.org/doc/html/rfc7908, 2016. RFC 7908, IETF, Accessed: 2025-08-22.

[29] Sean Turner and Oliver Borchert. BGPsec Algorithms, Key Formats, and Signature Formats. RFC 8608, 2019. URL: https://www.rfc-editor.org/rfc/rfc8608, doi:10.17487/RFC8608.

[30] University of Oregon. University of oregon route views project, 2025. Accessed: 2025-08-26. URL: http://www.routeviews.org/.

[31] US–China Economic and Security Review Commission. 2010 report to congress of the u.s.-china economic and security review commission. Technical report, U.S. Government Printing Office, November 2010. Accessed: 2025-08-27. URL: https://www.uscc.gov/annual-report/2010-annual-report-congress.

[32] Zheng Zhang, Z. Morley Mao, Ming Zhang, Bo Gao, Ben Y. Zhao, and Anthony D. Joseph. Practical defenses against bgp prefix hijacking, 2007. Proceedings of ACM CoNEXT 2007, Accessed: 2025-08-22. URL: https://web.eecs.umich.edu/~zmao/Papers/conextDefendHijack07.pdf.

Ringraziamenti

Grazie a tutti