Protocolos de enrutamiento anónimos para redes Ad-Hoc

2ª Parte

María Mercedes Rodríguez García

DISEÑO AVANZADO DE REDES

MÁSTER EN INVESTIGACIÓN EN INGENIERÍA DE SISTEMAS Y DE LA COMPUTACIÓN

SDAR
(an efficient Secure Distributed Anonymous Routing protocol for mobile and wireless ad-hoc networks)
2004

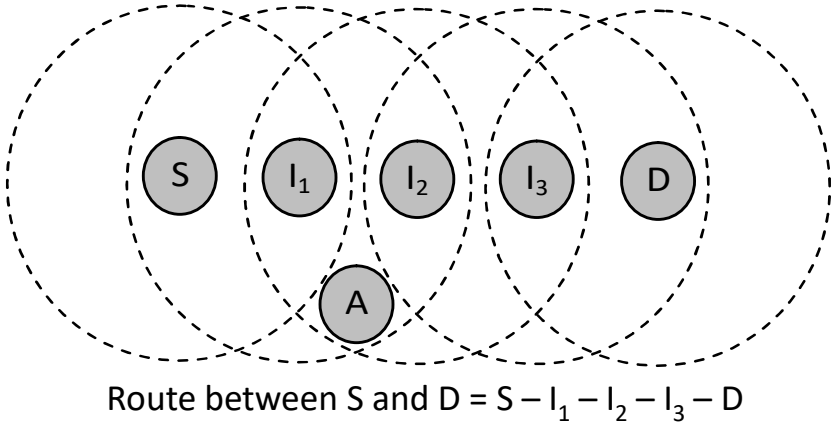
-En estas diapositivas se expone una **versión simplificada** de SDAR utilizada en todas las comparativas de protocolos de enrutamiento posteriores, libros y surveys -

- Hay elementos que no quedan bien justificados y explicados en el trabajo original -

-En estas diapositivas se realizan determinados supuestos (sombreados en amarillo) por no quedar clarificados en el trabajo original -

SDAR

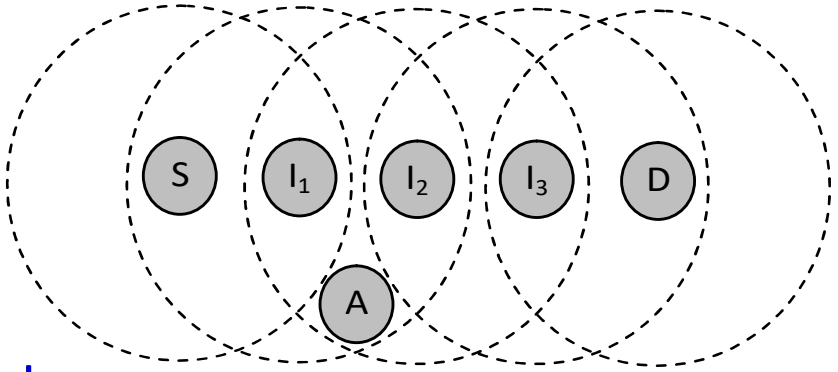
Fase **RREQ**:



- Fase iniciada por el nodo origen S
- Broadcast (inundación)
- <RREQ, PK_{temp}, $trapdoor_D$, padding, $path_S$>
  - PK_{temp} : clave pública de un solo uso (≡ clave pública de sesión) creada por S y empleada para encriptar el path de los nodos intermedios. También utilizada como número de secuencia.
  - $trapdoor_D = E_{PK_D}(ID_D, K_S, length\_padding)$
    - PK_D: clave pública del nodo destino
    - ID_D: identidad del nodo destino
    - K_S: clave de sesión que solo compartirá con D
  - padding: relleno para evitar m*essage size attacks*
  - $path_S = E_{K_S}(ID_S, SK_{temp}, N_S \ldots)$
    - N_S : nonce utilizado como índice para buscar la ruta en la tabla de rutas
    - SK_{temp} : clave privada correspondiente a PK_{temp}

SDAR

¿Qué hace un nodo cuando recibe un mensaje RREQ?



Route between S and D = S – I_1 – I_2 – I_3 – D

Por ejemplo, ¿qué haría el nodo $\mathbf{I_3}$ al recibir **<RREQ, PK_{temp}, $trapdoor_D$, padding, $path_S$, $path_{I1}$, $path_{I2}$>**?

1. Verificar si es la primera vez que recibe ese RREQ
2. Verificar si es el destino ¿cómo? Intentado abrir el $trapdoor_D$
3. Si no es el destino,
   a. modificar el mensaje RREQ antes de difundirlo

      <RREQ, PK_{temp}, $trapdoor_D$, padding, $path_S$, $path_{I1}$, $path_{I2}$, $path_{I3}$>

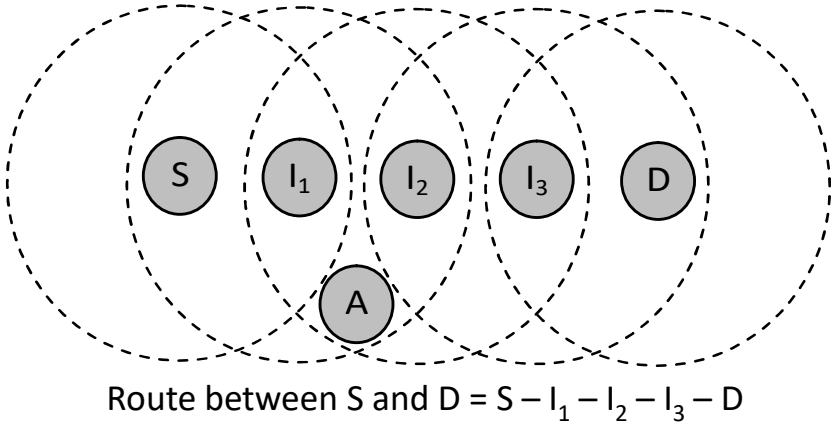
      $path_{I3} = E_{PK_{temp}}$ ($\mathbf{K_{I3}}$, $\mathbf{N_{I3}}$, …)

      K_{I3}: clave de sesión que solo compartirá con S y D

   b. registrar N_{I3} , K_{I3} y ID_{I2} en la tabla de rutas

	ID_{I2}	K_{I3}	N_{I3}	

SDAR



Route between S and D = S – I_1 – I_2 – I_3 – D

Fase **RREP**:

- Fase iniciada por el nodo destino D

- El nodo destino D:

  a. desencripta el *trapdoor* y el *path* de cada nodo

     <RREQ, PK_{temp}, $trapdoor_D$, padding, $path_S$, $path_{I1}$, $path_{I2}$, $path_{I3}$>

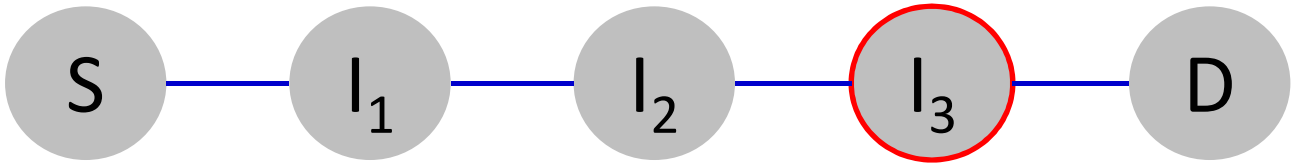
  b. registra K_{I3}, K_{I2}, K_{I1}, K_S, N_{I3}, N_{I2}, N_{I1}, N_S en su tabla de rutas

  c. construye el mensaje RREP: <RREP, N_{I3}, $onion_{I3}$>

     $onion_{I3} = E_{K_{I3}} (N_{I2}, E_{K_{I2}} (N_{I1}, E_{K_{I1}} (N_S, E_{K_S} (K_{I3}, K_{I2}, K_{I1}, N_{I3}, N_{I2}, N_{I1}, \ldots))))$

  d. envía el mensaje RREP mediante unicast al nodo ID_{I3}

SDAR



¿Qué hace un nodo cuando
recibe un mensaje RREP?
Por ejemplo, ¿qué haría el nodo $\mathbf{I_3}$ al recibir **<RREP, N_{I3}, $onion_{I3}$>** donde
$onion_{I3} = E_{K_{I3}} (N_{I2}, E_{K_{I2}} (N_{I1}, E_{K_{I1}} (N_S, E_{K_S} (K_{I3}, K_{I2}, K_{I1}, N_{I3}, N_{I2}, N_{I1}, \ldots))))$?

1. localizar la entrada correspondiente a N_{I3} en la tabla de rutas.

	ID_{I2}	K_{I3}	N_{I3}	

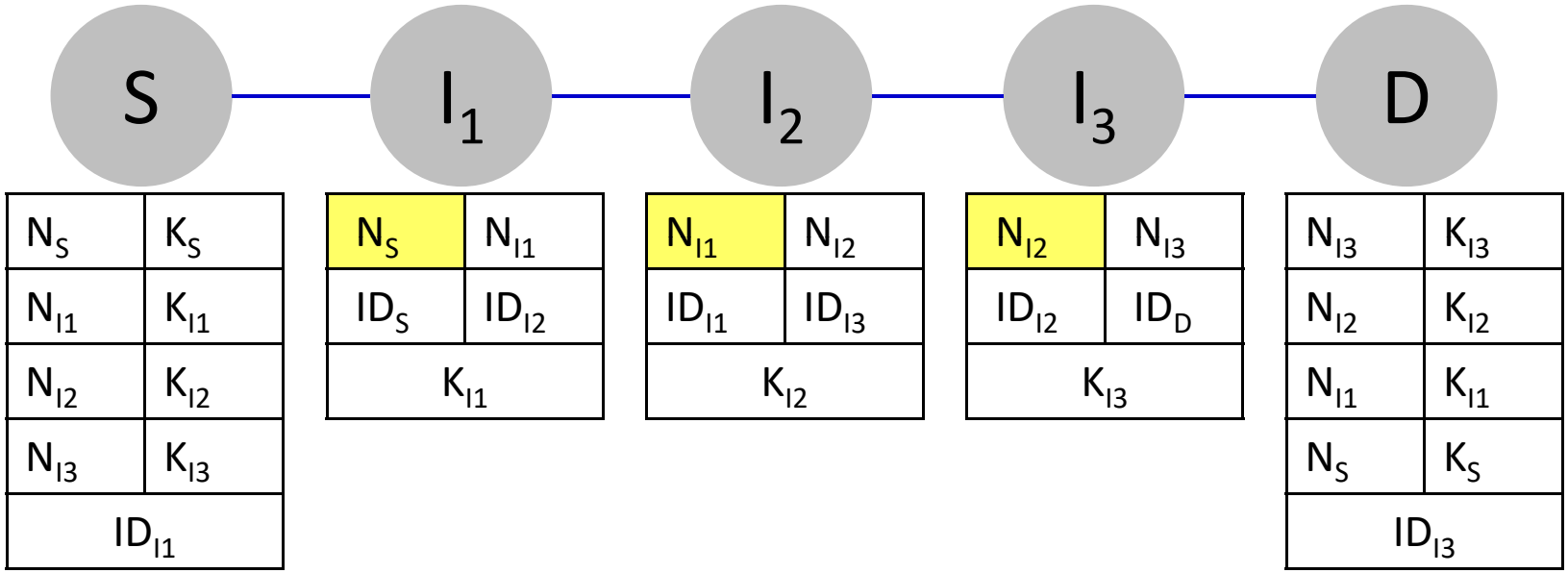
2. abrir su capa onion $E_{K_{I3}} (\mathbf{N_{I2}}, E_{K_{I2}} (N_{I1}, E_{K_{I1}} (N_S, E_{K_S} (K_{I3}, K_{I2}, K_{I1}, N_{I3}, N_{I2}, N_{I1}, \ldots))))$
3. registrar ID_D en la tabla de rutas

N_{I2}	ID_{I2}	K_{I3}	N_{I3}	**ID_D**

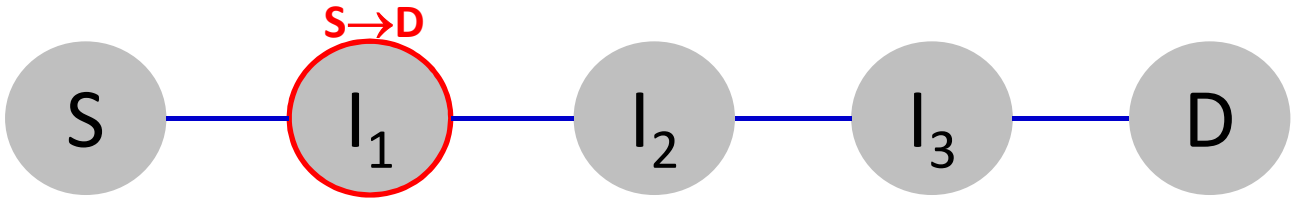
4. modificar el mensaje RREP antes de dirigirlo a I_{I2}.

<RREP, N_{I2} , $onion_{I2}$> $onion_{I2} = E_{K_{I2}} (N_{I1}, E_{K_{I1}} (N_S, E_{K_S} (K_{I3}, K_{I2}, K_{I1}, N_{I3}, N_{I2}, N_{I1}, \ldots)))$

SDAR


S
I_1
I_2
I_3
D
N_S K_S
N_{I1} K_{I1}
N_{I2} K_{I2}
N_{I3} K_{I3}
ID_{I1}
N_S N_{I1}
ID_S ID_{I2}
K_{I1}
N_{I1} N_{I2}
ID_{I1} ID_{I3}
K_{I2}
N_{I2} N_{I3}
ID_{I2} ID_D
K_{I3}
N_{I3} K_{I3}
N_{I2} K_{I2}
N_{I1} K_{I1}
N_S K_S
ID_{I3}

SDAR



Fase **DATA**:

- Fase iniciada por el nodo origen S
- Bidireccional: S→D, D→S
- Unicast

¿Qué hace un nodo cuando recibe un mensaje DATA?
Por ejemplo, ¿qué haría el nodo $\mathbf{I_1}$ al recibir **<DATA, N_S, onion-data$_{I1}$>** donde onion-data$_{I1}$= $E_{K_{I1}}(E_{K_{I2}}(E_{K_{I3}}(E_{K_S}(data))))$?

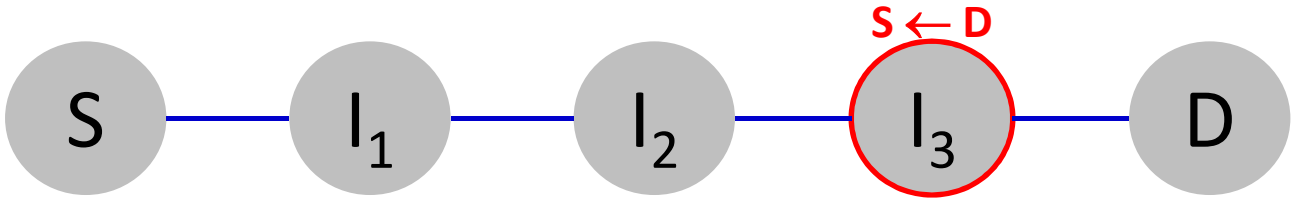
1. localizar la entrada correspondiente a N_S en la tabla de rutas

N_S	ID_S	K_{I1}	N_{I1}	ID_{I2}

2. modificar el mensaje DATA antes de dirigirlo a I_2

   <DATA, N_{I1}, onion-data$_{I2}$> onion-data$_{I2}$ = $E_{K_{I2}}(E_{K_{I3}}(E_{K_S}(data)))$

SDAR



Fase **DATA**:

¿Qué hace un nodo cuando recibe un mensaje DATA?
Por ejemplo, ¿qué haría el nodo $\mathbf{I_3}$ al recibir **<DATA, N_{I3}, onion-data$_{I3}$>** donde onion-data$_{I3}$ = $E_{K_{I3}}(E_{K_{I2}}(E_{K_{I1}}(E_{K_S}(data))))$?

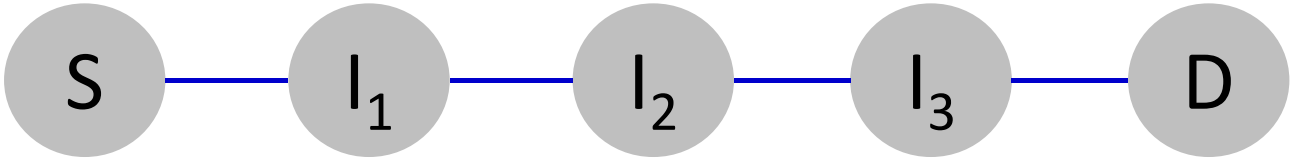
1. localizar la entrada correspondiente a N_{I3} en la tabla de rutas

N_{I2}	ID_{I2}	K_{I3}	N_{I3}	ID_D

2. modificar el mensaje DATA antes de dirigirlo a I_2

   <DATA, N_{I2}, onion-data$_{I2}$> onion-data$_{I2}$ = $E_{K_{I2}}(E_{K_{I1}}(E_{K_S}(data)))$

SDAR



Justificar uso de:

1. PK_{temp}, SK_{temp}
2. PK_D, SK_D
3. K_S
4. Padding

Inconvenientes:

1. Trapdoor con criptografía asimétrica