Presentación de
“**Implementación de cifrado broadcast para mensajes cortos en wifi**”.

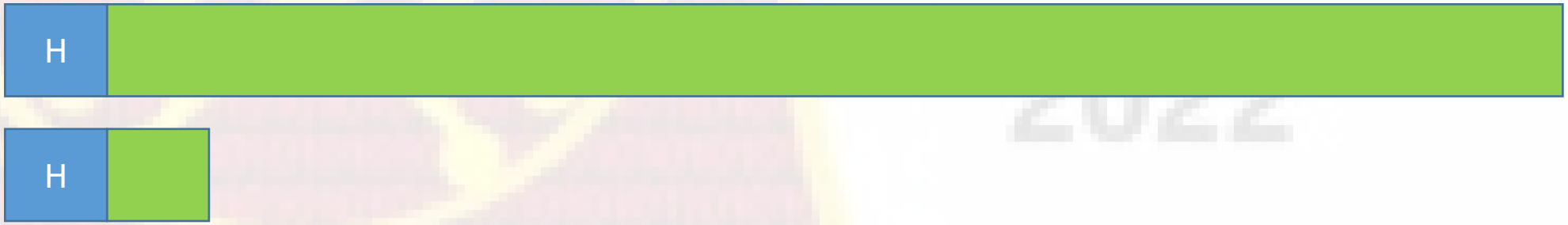
José Luis Salazar, Julián Fernández-Navajas, José Ruiz-Mas y Guillermo Azuara .

Grupo CeNIT.
Instituto de Investigación en Ingeniería de Aragón (I3A).
Universidad de Zaragoza

- Introducción
- Cifrado broadcast para mensajes cortos
- Soluciones propuestas
- Conclusiones

Ineficiencia paquetes pequeños

- Uso creciente servicios en tiempo real
- Paquetes muy pequeños enviados de forma muy frecuente
- Encabezados de seguridad añaden ineficiencia
- Ejemplo: Paquetes pequeños sobre 802.11



Ineficiencia paquetes pequeños

- **Soluciones**
  - **Agregación** de tramas 802.11
  - Encabezado de seguridad varios paquetes pequeños
  - Uso de paquetes broadcast para mejorar eficiencia (escenarios inalámbricos)

Equilibrio seguridad - eficiencia

- Agregar paquetes cifrados pequeños
- Asegurar cada paquete solo puede ser descifrado por receptor
- **Sistema clásico**: Televisión por Cable: u_i encabezado de cada usuario, H*dr* campo común y c_i mensaje cifrado de cada usuario

$$(u_1, u_2, \dots, u_n, Hdr, c_1, c_2, \dots, c_n)$$

- **Sistema propuesto**: se fusionan todos los encabezados individuales en un campo común H*dr'*

$$(Hdr', c_1, c_2, \dots, c_n)$$

Equilibrio seguridad - eficiencia

- **Sistema propuesto**: se fusionan todos los encabezados individuales en un campo común H*dr'*

$$(Hdr', c_1, c_2, \ldots, c_n)$$

  - Se reduce la cantidad total de información transmitida (**ahorro ancho de banda**)
  - Sólo el destinatario podrá descifrar información (utilizando H*dr' y* su clave privada)

Equilibrio seguridad - eficiencia

- **Puesta a punto** para explotación en **wifi**
  - Posibles otras funciones intermedias alternativas
  - Mejoras enfocadas en dos **problemas de implementación**:
    - **Tamaño** de los **números primos** (tamaño clave privada)
    - **Eficiencia** de la **generación** de la **semilla** aleatoria

• Modelo de cifrado broadcast

- Modelo propuesto en por Baudron, Pointcheval y Stern: “Extended notions of security for multicast public key cryptosystems” (2000)
- Consta de **cuatro algoritmos** probabilísticos:
  - **Configuración (λ)**: genera parámetros globales y devuelve $\boldsymbol{EK}$ (clave secreta)
  - **Extracción** $(\boldsymbol{i}, \boldsymbol{EK})$: genera claves privadas de usuario (p_i, xi)
  - **Cifrado** $(\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{un}, \boldsymbol{EK}, \boldsymbol{m_1}, \boldsymbol{m_2}, \ldots, \boldsymbol{mn})$: produce (Hdr, ET)
  - **Descifrado** $(\boldsymbol{Hdr}, \boldsymbol{ET}, \boldsymbol{i}, \boldsymbol{pi}, \boldsymbol{xi})$: devuelve m_i para i

• Modelo de cifrado broadcast para mensajes cortos

- Adaptación a mensajes cortos: SMMBE ”Short Message Multichannel Broadcast Encryption” (2019), mediante criptografía simétrica pero utilizando aritmética modular (Teorema Chino de los Restos).
- Criptografía simétrica más lenta, pero compensada por la corta longitud de los mensajes
- Consta de **cuatro algoritmos**:
  - Configuración (λ)
  - Extracción
  - Cifrado $(u_1, u_2, \dots, un, EK, m_1, m_2, \dots, mn)$
  - Descifrado (Hdr, ET, i, pi, xi)

• Modelo de cifrado broadcast para mensajes cortos

- **Configuración (λ)**
  - Entrada λ
  - Se eligen n primos p_i (uno por usuario) y n enteros aleatorios $x_i \in Z_{p_i}^*$, tales que $\text{mcd}(x_i, pi - 1) = 1$
  - Se define $N = \prod_{i=1}^{n} p_i$ y $EK = ((p_1, x_1), (p_2, x_2), \ldots, (p_n, xn))$, cada par es la clave secreta de descifrado de cuyo envío se encargará el algoritmo **Extracción.**

• Modelo de cifrado broadcast para mensajes cortos

- **Cifrado** $(\boldsymbol{u_1, u_2, \dots, un, EK, m_1, m_2, \dots, mn})$
  - Elige un entero aleatorio $Hdr \overset{\$}{\leftarrow} \mathrm{Z}^*_{\min_j\{pj\}}$
  - Define $Hdr_i = min\{g \geq Hdr \quad \backslash \mathrm{g}\ es\ un\ generador\ de\ \mathrm{Z}^*_{p_i}\}$
  - Calcula $ET = \sum_{i=1}^{n}\left(m_i + Hdr_i^{x_i}\ (mod\ pi)\right) \cdot \frac{N}{p_i}\left[\left(\frac{N}{p_i}\right)^{-1}\ (mod\ pi)\right](mod\ N)$
  - Salida (Hdr, ET)

- # Modelo de cifrado broadcast para mensajes cortos
  - **Descifrado** $(\boldsymbol{Hdr}, \boldsymbol{ET}, \boldsymbol{i}, \boldsymbol{pi}, \boldsymbol{xi})$
    - Calcula $m_i = (ET - Hdr_i^{x_i})(mod\ pi)$
    - Consecuencia Teorema Chino de los Restos

Entorno de aplicación: WiFi

- Necesidad de enviar paquetes pequeños → *overhead*
- Limitaciones en la espera para agrupar por la QoS
- Limitación en el tamaño de las claves y en su generación
- Equilibrio eficiencia – seguridad
  - Algoritmos eficientes si tamaño de paquetes de usuario tienen límite superior
  - Ganancias óptimas si todos los paquetes son del mismo tamaño (Ej. VoIP) → seleccionar tamaño de clave óptimo

Entorno de aplicación: WiFi

- Enlace inalámbrico, nuevos retardos:
  - Acceso al medio
- Usar tramas multicast en vez de unicast mejorará la eficiencia
- 802.11n permite agregación:
  - A-MPDU (Aggregated MAC Protocol Data Unit)
  - A-MSDU (Aggregated MAC Service Data Unit)
- Escenario adecuado para implantación del sistema propuesto (mejor A-MSDU)
- Mejora la seguridad porque cada MSDU será inaccesible para los que no sean receptores

Usuarios y cabecera mismo rol. Mensajes a cifrar: los diferentes MSDUs

Problemas de implementación

- Conflicto de intereses: seguridad vs eficiencia
- Cálculo de Hdr' (herramienta función pseudoaleatoria) marca cota inferior de seguridad
- Cálculo de generadores de anillos requiere gran potencia de cálculo en los nodos
- A **mayor tamaño de clave → mayor seguridad** pero peor eficiencia, por:
  - Incremento muy notable del **tamaño del mensaje** a transmitir frente al original
  - Aumentar tamaño de los elementos en el anillo Z_p eleva el **tiempo de cálculo**

Ajuste del tamaño de la clave

- **Fijar el tamaño de los primos p_i a una longitud fija** → Establecer λ
  - Por ejemplo a 1024 bits (se podría aumentar si fuera necesario)
  - Podríamos visualizar transmisión de cada MSDU como un canal de seguridad de 1024 bits
  - Limitar la política de claves:
    - ¿Qué hacemos si llegan paquetes mayores?
    - ¿De media cuántos paquetes vacíos cifro?

Ajuste del tamaño de la clave

- **¿Qué hacemos si llegan paquetes mayores que el tamaño de la clave?**
  - Canales necesarios en función del tamaño máximo del mensaje: $j = \left\lceil \frac{m_i}{\lambda} \right\rceil$ y definimos $N_i = \prod_{k=1}^{j} p_{ij}$, donde p_{ij} será el primo asignado al usuario i para su canal de seguridad j
  - Esto implica un pequeño cambio en la elección de los parámetros y en la función de cifrado:
    - Claves de usuario i: $\left((p_{i1}, \dots, p_{ij}), xij = \prod_{k=1}^{j} x_{ik}\right)$, donde $x_{ij} \in \mathrm{Z}_{N_i}^{*}$ es el producto de las claves secretas que se deberían asignar a los canales de seguridad.
    - Podemos definir: $m_{ij} = m_i + Hdr_i^{x_{ij}} \ (mod\ Ni)$, por tanto si en esos n canales hay l usuarios, se reescribe la fórmula de cifrado como:

$$ET = \sum_{i=1}^{l} \left(\sum_{k=1}^{j} \left(m_{ij}(mod\ pik) \cdot \frac{N}{p_{ik}} \left[\left(\frac{N}{p_{ik}} \right)^{-1} (mod\ pik) \right] \right) \right) (mod\ N)$$

Ajuste del tamaño de la clave

- **¿Qué hacemos si llegan paquetes mayores que el tamaño de la clave?**
  - Podemos definir: $m_{ij} = m_i + Hdr_i^{x_{ij}} \ (modNi)$, por tanto si en esos n canales hay l usuarios, se reescribe la fórmula de cifrado como:

$$ET = \sum_{i=1}^{l} \left(\sum_{k=1}^{j} \left(m_{ij}(mod\ pik) \cdot \frac{N}{p_{ik}} \left[\left(\frac{N}{p_{ik}} \right)^{-1} (mod\ pik) \right] \right) \right) (mod\ N)$$

  - Función de descifrado (ahorro reconstrucción todo MSDU):

$$m_i = \left(ET - Hdr_i^{x_{ij}} \right) (modN_i)$$

Ajuste del tamaño de la clave

- **¿De media cuántos paquetes vacíos cifro?**
  - Depende de la distribución probabilística de la variable aleatoria que modela la longitud del mensaje
  - Si suponemos una media M y la capacidad de los canales C, la ratio de espacio desaprovechado en cada MSDU, será:

$$\frac{M(mod\ C)}{\left\lceil \frac{M}{C} \right\rceil}$$

  - Muy importante conocer la distribución de la longitud de los MSDU para **minimizar el numerador y así mejorar eficiencia**.

Generador aleatorio

- Inicialmente semilla aleatoria común Hdr de la que luego cada usuario calcula Hdr_i
- Suficiente con modelar oráculo aleatorio y dejar al usuario la elección de la función
- Cálculo de generadores en anillo es costoso (ámbito comunicaciones en tiempo real)
- **Se busca un algoritmo que genere diferentes salidas aleatorias para cada usuario, a partir de semilla fija y consumiendo pocos recursos.**
- **Función HMAC**
  - En función del tamaño necesario se pueden encadenar varias salidas
  - A elección del desarrollador
- **Renovación de las claves (p_i,x_i) más frecuente que si se usa PKC.**
  - Posible solución off-line: almacén claves iguales uso sincronizado, mismo método de encadenamiento que HMAC.

Conclusiones

- **Puesta en práctica** algoritmo envío mensajes cortos cifrados en broadcast o multidifusión
- Entorno **wifi**
- **Mejorado ajuste de longitud de las claves**
  - Canales de seguridad de tamaño fijo
  - Cada canal claves secretas compartidas por origen y destino
  - Cada usuario conjunto de canales (conocerá sus claves)
- **Mejorada generación de números aleatorios**
  - Uso de HMACs como generadores pseudoaleatorios
  - Semilla compartida por todos los usuarios de la multidifusión
  - Operaciones básicas son calculadas con funciones hash en lugar de operaciones de aritmética finita (entre ellas exponenciaciones)
  - Implementación beta realizada con éxito (Python)

Muchas gracias

Esta publicación ha sido financiada en parte por la ayuda T31 _20R al grupo CeNIT de la Universidad de Zaragoza, financiada por el Gobierno de Aragón.