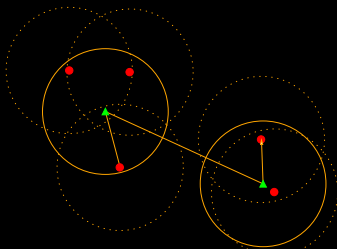


Fraud within Asymmetric Multi-Hop Cellular Networks

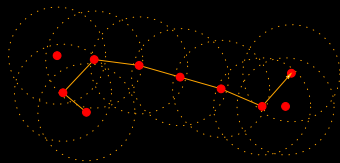
Gildas Avoine

EPFL, Lausanne, Switzerland

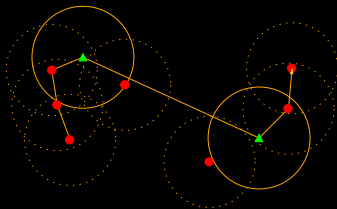




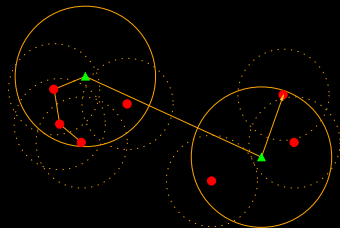
Single-hop cellular network



Multi-hop network



Multi-hop cellular network



Asymmetric multi-hop cellular network

Problem: How to encourage the mobile stations (**nodes**) to relay packets for the benefit of other nodes?

Solution: Rewarding the nodes on the packet path (**payment**).

Proposition: Jakobsson, Hubaux, Buttyán, FC'03

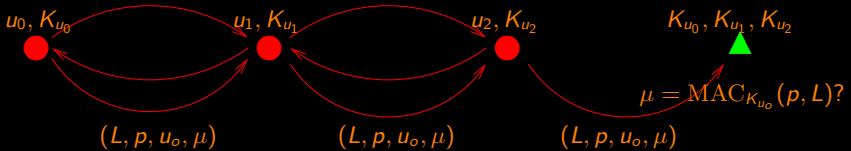
Description of the scheme

Communicating freely in a cell

Recovering secret keys using a side channel attack

Description of the scheme

- Very **lightweight** scheme.
- Small cheating are **possible**. However large cheating are **detected** and misbehaviors are punished.
- The originators are charged for the packets they send.
- The intermediaries are rewarded **probabilistically**.
- The operator audits the traffic to detect misbehaviors (statistical analysis).

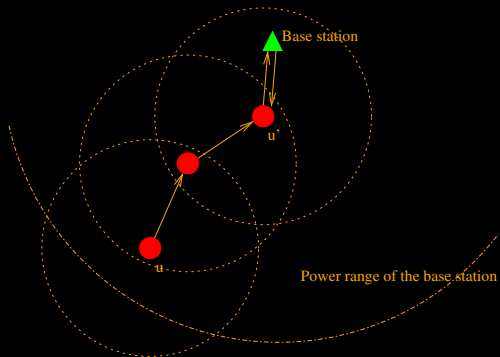


1. Each user registers to his operator in order to obtain an identity u and a symmetric key K_u .
1. If the base station can be reached in a single hop...
2. u selects a neighbor (routing protocol).
3. u sends a **forward request** to u_i containing a reward level L .
4. If u_i agrees to forward, he sends an **ack** to u .
6. u sends (L, p, u_o, μ) to u_i , where $\mu = \text{MAC}_{K_{u_o}}(p, L)$.
7. The base station checks the MAC with the stored key K_{u_o} .

1. Each user u verifies whether $d_{\mathcal{H}}(\mu, K_u) \leq h$. If the test succeeds, we say that the user has a **winning ticket**. In this case, he can claim a reward for his forwarding.
2. He records $(u_{\text{prev}}, u_{\text{next}}, \mu, L)$ in a list M .
3. He sends M to the operator from time to time.

Communicating freely in a cell

- This attack is a **fake identity-based attack** which consists for two attackers in communicating freely in a cell.
- If a user u sends a message to a user u' who is not in his neighborhood, then the packet is sent to the base station through some other users.
- If u' is on the path from u to the base station, he **should not keep** the packet but **should wait** that it comes back from the base station.



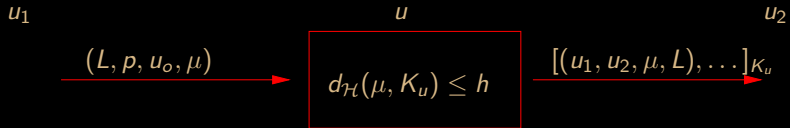
→ The problem is due to the fact that there is **no authentication** between the nodes.

→ Force the packet to pass through the base station in order to be **useful** for the recipient: the nodes should **encrypt** the packets they forward s.t. only the base station is able to decrypt them.

→ A node could encrypt the packet with a given probability δ . So, the probability that the attack succeeds is $(1 - \delta)^n$ where n is the number of intermediaries on the packet path.

E.g. with $n = 5$ and $\delta = \frac{1}{2}$, the probability of success is about 3%.

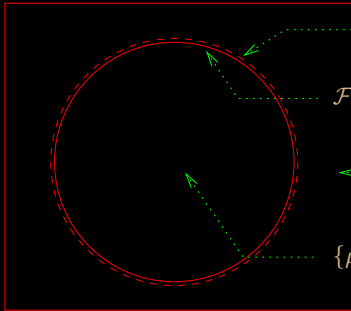
Recovering secret keys using a side channel attack





Let $K \in \{0, 1\}^\ell$ be the secret key to recover, where $\ell \in \mathbb{N}^*$; let h be a public value such that $0 < h < \ell$. If we have an oracle \mathcal{O} such that for a request $\mu \in \{0, 1\}^\ell$, \mathcal{O} returns **true** if and only if $d_{\mathcal{H}}(\mu, K) \leq h$, then we can recover K with a complexity in terms of requests to the **oracle** in

$$\frac{2^{\ell+1}\ell(h+1)}{(2h+1)(\ell-h)\binom{\ell}{h}} + \ell.$$



$$\mathcal{F}_{\text{out}} := \{\mu \in \{0, 1\}^\ell \mid d_{\mathcal{H}}(\mu, K) = h + 1\}$$

$$\mathcal{F}_{\text{in}} := \{\mu \in \{0, 1\}^\ell \mid d_{\mathcal{H}}(\mu, K) = h\}$$

$$\{\mu \in \{0, 1\}^\ell \mid d_{\mathcal{H}}(\mu, K) > h\}$$

$$\{\mu \in \{0, 1\}^\ell \mid d_{\mathcal{H}}(\mu, K) < h\}$$

Step 1: Finding μ on one of the frontiers: $\frac{2^{\ell+1}\ell(h+1)}{(2h+1)(\ell-h)\binom{\ell}{h}}$ requests.

Step 2: Recovering K from μ : ℓ requests.

Las Vegas algorithm: (LV)

- Pick (randomly) $\mu \in \{0, 1\}^\ell$ and $i \in [1, \ell]$
- Send μ and $\mu^{(i)}$ to the oracle \mathcal{O}
- If $\mathcal{O}(\mu) \neq \mathcal{O}(\mu^{(i)})$
Then return “ μ is on the frontiers”
Else return \perp

Where $\mu^{(i)}$ is the new value of μ when the i th bit is flipped.

Probability that $\mu \in \mathcal{F}_{\text{in}}$ is $\frac{\binom{\ell}{h}}{2^\ell}$.

Probability that LV answers given $\mu \in \mathcal{F}_{\text{in}}$ is $(1 - \frac{h}{\ell})$.

Probability that $\mu \in \mathcal{F}_{\text{out}}$ is $\frac{\binom{\ell}{h+1}}{2^\ell}$.

Probability that LV answers given $\mu \in \mathcal{F}_{\text{out}}$ is $\frac{h}{\ell}$.

Finally, the probability that the LV algorithm answers is

$$\xi := \frac{1}{2^\ell} \left[\binom{\ell}{h} \left(1 - \frac{h}{\ell}\right) + \binom{\ell}{h+1} \left(\frac{h}{\ell}\right) \right].$$

We define a **Monte Carlo** algorithm (MC) from the **Las Vegas** algorithm . Let C be the number of rounds of MC in order to find a value μ on the frontier; we have:

$$\text{Prob}(C = c) = \xi(1 - \xi)^{c-1}.$$

It follows that the average number of rounds is

$$\frac{1}{\xi} = \frac{2^\ell}{\binom{\ell}{h} \left(1 - \frac{h}{\ell}\right) + \binom{\ell}{h+1} \frac{h}{\ell}}.$$

Assume that μ is on the frontier (w.l.o.g. we choose $\mu \in \mathcal{F}_{\text{in}}$).

The attacker flips **independently** every bit of μ and sends these values $\mu^{(i)}$ to the oracle.

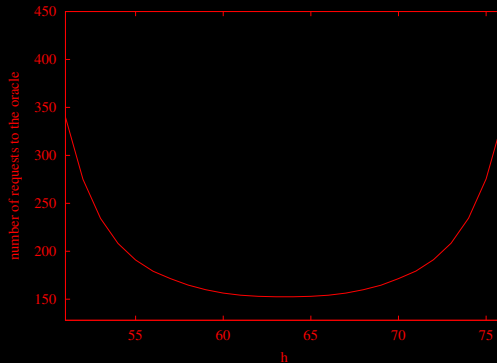
If $\mathcal{O}(\mu^{(i)})$ Then $\mu_i \neq K_i$ Else $\mu_i = K_i$.

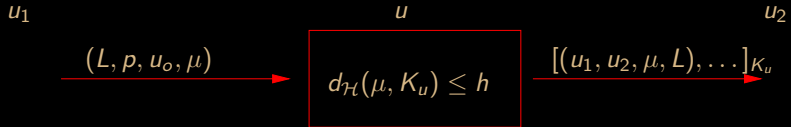
Complexity of Step 2 in terms of calls to the oracle is therefore ℓ .

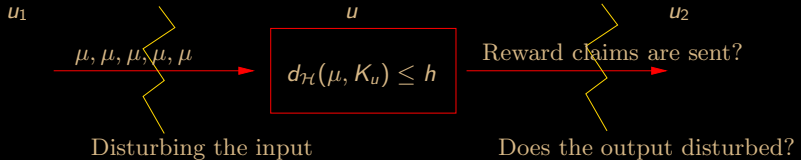
The complexity of the full attack is

$$\frac{2}{\xi} + \ell = \frac{2^{\ell+1}\ell(h+1)}{(2h+1)(\ell-h)\binom{\ell}{h}} + \ell.$$

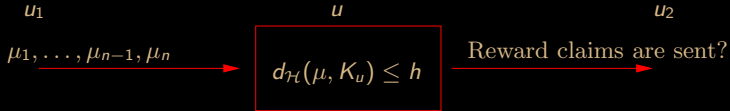








Naive idea: the user may reject beams of equal random values



The $n - 1$ values fill the buffer, except the last space which is used to check μ .

Description of the scheme

Communicating freely in a cell

Recovering secret keys using a side channel attack

End