CS261

Scribe: Raluca Sauciuc

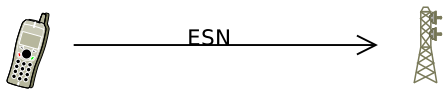
November 25, 2008

Security is a systems problem. The easiest route is not to attack crypto, but bypass it altogether. Paper exemplifies this within the banking industry. To summarize:

- "crypto is not magic pixie dust"
- Roger Needham: "if you think crypto will solve your security problem, you don't understand crypto and you don't understand your problem."

1 Cellphone Security

AT & T invented them in the late 70's. First generation:



Each cellphone has a 32-bit ID (Electronic Serial Number) which is sent to the base station. Telco has a DB mapping ID's to accounts, so they know who to bill. Attacks:

- cellphone cloning
- trivial to eavesdrop (scanners)

Cloning was a big deal - you could hack your own phone and change the ID to a random number. The base station would allow calls anyway while checking with the DB in the background (waiting for the DB check was considered too expensive to perform synchronously before the call). If the ID turned out to be invalid, there was no mechanism for tearing down the connection. There was a very popular scheme for selling cheap international calls by performing a 3-way call: first call would be kept alive for 12h (the maximum time, since the base station couldn't tear down this call with an invalid ID), while the second one dials international numbers on demand. The companies' response was to keep cached blacklists to detect invalid IDs. But blacklists had to be propagated fast

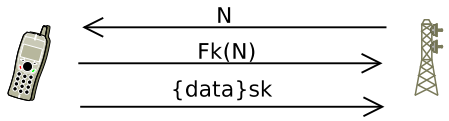
enough between base stations, and even then the first fraudulent call would go through.

An improvement to this basic cloning was tumbling: the phone picks a new random ESN each time. The companies upgraded to tear down calls in progress. The next attack was to use valid ESNs. The only problem was to not set off the fraud alarms (otherwise use some ESNs for just a month, until the phone bill comes out).

Eavesdropping was very easy (you could even use old TVs for that!). Preferred places were airports or highways (just sit in the car and listen...)

Companies reportedly lost $500 mil. annually to fraud. But they already had the infrastructure in place, so they couldn't afford a complete redesign (changing all base stations), they had to stay backwards-compatible. Their fix was to switch to digital + add security mechanisms.

1.1 Digital



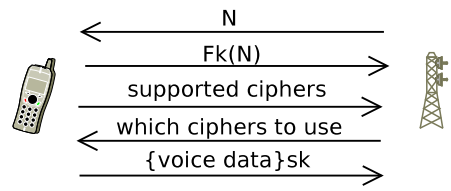
The phone has a 64-bit key K, shared with the phone company. A session key s_k of 500 random bits, constructed by $s_k = F'_k(N)$, is used as a many-time pad to encrypt the data (XOR).

But silence frames would be like known plain text and would allow the recovery of the session key. Therefore, all phones had an all 0's key installed as a default key. This would be the long lifetime key, with no entropy. A second, medium lifetime key would be generated and evolved from all nonces seen, and this key would be used for encryption. This way, you would need to see all nonces to intercept a call – otherwise the medium lifetime key is random enough.

1.2 GSM

Europe also had export control issues for crypto, just as US. The GSM standard defines 3 levels of security, in increasing order of security against eavesdropping:

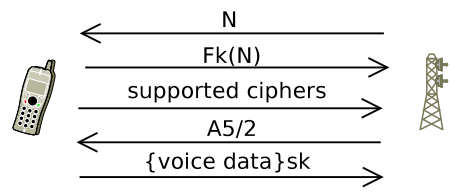
- A5/0 – no security at all
- A5/2 – 40-bit key, eliminates ordinary eavesdroppers
- A5/1 – full strength crypto algorithm



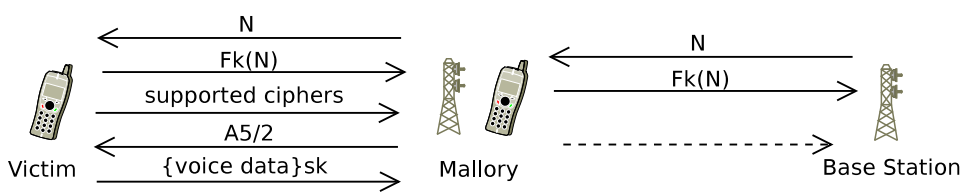
The session key s_k is constructed similarly ($s_k = F'_k(N)$). The difference is that now a list of supported ciphers is sent to the base station, which selects which cipher to use.

- Authentication algorithms have to be full-strength, otherwise money is lost due to fake calls / cloning fraud
- Voice confidentiality can be flaky

The "you're now in France" attack: you set up a fake base station and claim to only support A5/2 (no strong crypto allowed). Repeat the nonce N for a previously-recorded call that you wish to decrypt, so the session key stays the same. Then you crack the crypto and reconstruct the session key:



Man-in-the-middle attack:



Mallory breaks A5/2 and recovers the session key s_k, then places fake calls over the authenticated base station.

New versions of GSM fix this by choosing $s_k = F'_k(N, which-cipher-you-use)$

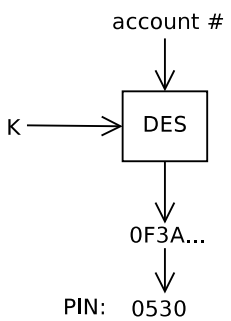
To summarize, these attacks are kind of rare – you need to fake base stations, etc. The easy path for fraud is more into the economic landscape: identity theft or prepaid cards with fraudulent (stolen) cards.

2 Banks & ATMs

1. First version, magnetic stripe has $(account\#, DES_k(PIN))$.

   But DES is breakable. Plus, there's no connection between the account number and the PIN, so you can change $DES_k(PIN)$ and put your encrypted PIN. Receipts also listed the full account number, which made this attack easy.

2. The magnetic stripe only has $(account\#)$. The PIN is derived from the account number: $PIN = F_K(account\#)$. After the DES encryption, the PIN is derived from the first four digits, by choosing for instance a modulo 10 representation.



   This scheme makes some numbers more likely to show up than others (frequency of digits is not the same). Plus, since the PIN number is assigned it's more likely to be written up somewhere; if not, the attacker can just snoop over your shoulder and see the PIN, then get the account # from the receipt.

3. The magnetic stripe has $(account\#, offset)$, with $PIN = offset + F_K(PIN)$. Now you can use your own PIN and the bank will choose appropriate offset.