Cryptanalysis of Crypto-1

Karsten Nohl
University of Virginia
nohl@virginia.edu

The secret cipher that secures Mifare Classic RFID tags used in access control systems, subway tickets, and various other security-related applications has recently been disclosed [1]. Since the security of the Mifare cards partly relies on the secrecy of this algorithm, we concluded that the cards are too weak for all security-related applications since the algorithm can be found with modest effort. A report for the Dutch government that assesses the impact of our findings on a nationwide ticketing system in the Netherlands was released on February 29th [2]. The report confirms our findings, but asserts that systems will likely be secure for another two years since the attack is still costly. In the report, the attack is estimated to require $9,000 worth of hardware to break secrets keys in a matter of hours. We argue that this is a gross over-estimate and present an attack that recovers secret keys within minutes on a typical desktop PC or within seconds on an FPGA. Our attack exploits statistical weaknesses of the cipher.

The Crypto-1 cipher consists of a *linear feedback shift register* (LFSR) and a filter function, f(·), as shown in Figure 1. During the initialization, the secret 48-bit key is loaded into the shift register and the string (ID xor R_b) is shifted into the state, where ID is the identifier of the tag, and R_b is a random number chosen by the tag. R_b is also sent to the reader as a first challenge in a challenge-response protocol in which tag and reader prove knowledge of the secret key. Since in our attack, the attacker only needs to communicate with the reader, the challenge can freely be chosen and does not need to be random.

In each clock cycle, the filter function, f(·), computes one bit of key stream from 20 LFSR bits. The function is composed from 6 instantiations of 3 smaller functions as depicted in Figure 2. These functions are statistically biased: if one input bit is held constant, then the output is '1' (or '0') more than 50% of the time. To compute this bias for a function with *n* inputs, we test its output distribution for all combinations of *n*-1 of the inputs while keeping the remaining input constant. For some of the inputs, the biases are as significant as 10/16 for f_c and 5/8 for f_a and f_b. This statistical weakness enables our attack.

To find bits of the key, we send challenges to the reader and analyze the first bit of key stream sent back by the reader. In these challenges, we hold constant the four inputs of one of the $f_{a/b}$ functions while changing some of the other bits in each challenge. To generate different challenges in which the chosen bits are constant we compute the impact that changes in other bits have on the LFSR feedback, which is trivial given knowledge of the LFSR structure

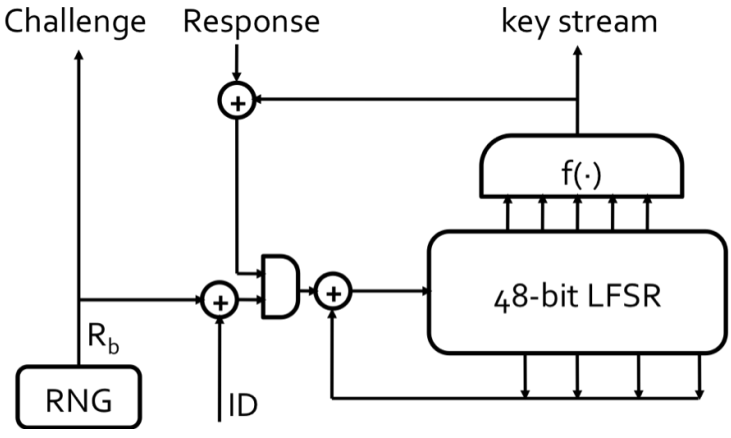

Figure 1. Crypto-1 stream cipher and initialization values.

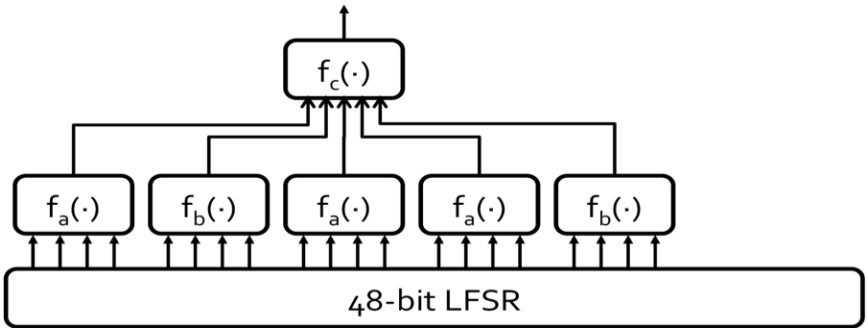

Figure 2. Structure of the Crypto-1 filter function.

and computationally cheap. The key stream bits we receive from the reader are biased towards either '0' or '1' because f_c is statistically biased, which lets us recover the output bit of the $f_{a/b}$ block under consideration. Knowing the output bit of the $f_{a/b}$ block we can then test for the inputs to the block. This process is repeated for a number of key bits until sufficiently many are known for a brute-force attack on the remaining unknown bits to become cheap. The number of challenges needed to recover key bits with high probability varies for different bits, but generally does not exceed a few dozens.

Using our technique, up to 32 key bits can be recovered, but some of the bits require significantly more challenges than others. We found particularly strong biases in 12 of the 32 bits that we can recover from the first bit of the key streams. After learning these 12 bits, all keys in the remaining 36-bit key space can be tried within 30 seconds on a single FPGA or within minutes on a typical PC.

Our attack assumes that we can recover the first bit of the key stream. This bit is combined with a random bit that is generated on the reader. We have shown previously that the random numbers are known to an attacker since they are generated deterministically [3]. The attacker needs knowledge of a single key that can, for example, be found using a brute-force attack on the entire key space. Once one key is known, an attacker can learn enough about a given reader to predict the random numbers it will generate, so any number of keys can be recovered using our statistical attack.

Our specific attack can be mitigated by using true random numbers on the reader. This does not fix the underlying weaknesses of the cipher, however, and more elaborate attacks will emerge. Instead of starting an arms-race around the security of a fundamentally flawed cipher, systems should upgrade to more secure cards that use publicly scrutinized cryptography.

[1] K. Nohl. *Lost Mifare obscurity raises concerns over security of OV-Chipkaart*. www.cs.virginia.edu/~kn5f/

[2] TNO. *Security Analysis of the Dutch OV-Chipkaart*. www.translink.nl/media/bijlagen/nieuws/TNO_ICT_-_Security_Analysis_OV-Chipkaart_-_public_report.pdf

[3] K. Nohl, H. Plötz. *Mifare—Little security despite Obscurity*. Talk at 24C3. events.ccc.de/congress/2007/Fahrplan/events/2378.en.html