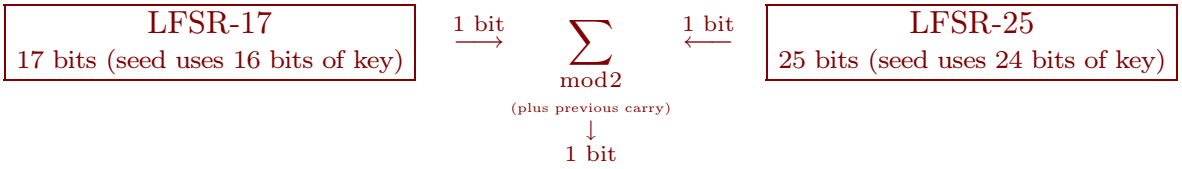
Example 63. (CSS) The CSS (content scramble system) is based on 2 LFSRs and used for the encryption of DVDs. Let us indicate (in a slightly oversimplified way) how to break it.

CSS was introduced in 1996 and first compromised in 1999. One big issue is that its key size is 40 bits. Since $2^{40} \approx 1.1 \cdot 10^{12}$ is small by modern standards, even a direct brute-force attack in time 2^{40} is possible.

However, we will see below that poor design makes it possible to attack it in time 2^{16}.

Historic comment. 40 bits was the maximum allowed by US export limitations at the time.

https://en.wikipedia.org/wiki/Export_of_cryptography_from_the_United_States



CSS PRG combines one 17-bit LFSR and one 25-bit LFSR. The bits output by the CSS PRG are the sum of the bits output by the two LFSRs (this is the usual sum, including carries).

The 40 bit key is used to seed the LFSRs (the 4th bit of each seed is "1", so we need $16+24=40$ other bits).

Here's how we break CSS in time 2^{16}:

- If a movie is encrypted using MPEG then we know the first few, say x (6-20), bytes of the plaintext.
- As in Example 57, this allows us to compute the first x bytes of the CSS keystream.
- We now go through all 2^{16} possibilities for the seed of LFSR-17. For each seed:
  - We generate x bytes using LFSR-17 and subtract these from the known CSS keystream.
  - This would be the output of LFSR-25. As in Example 62, we can actually easily tell if such an output could have been produced by LFSR-25. If yes, then we found (most likely) the correct seed of LFSR-17 and now also have the correct state of LFSR-25.

This kind of attack is known as a correlation attack.

https://en.wikipedia.org/wiki/Correlation_attack

Comment. Similar combinations of LFSRs are used in GSM encryption (A5/1,2, 3 LFSRs); Bluetooth (E0, 4 LFSRs). Due to certain details, these are broken or have serious weaknesses; so, of course, they shouldn't be used. However, it is difficult to update things implemented in hardware...

Sad but important lessons

Review. CSS (content scramble system) is based on 2 LFSRs whose outputs are added with carry (the carry is important because it combines the LFSRs in a nonlinear way).

Combining LFSRs in a nonlinear fashion is a good idea for constructing PRGs for cryptographic purposes (especially because they are simple to implement in hardware). However, as the examples of CSS as well as GSM/Bluetooth encryption show, a lot of attention has to be paid to the details in order not to compromise security.

CSS (and many other examples in recent history) teach us one important lesson:

Do not implement your own ideas for serious crypto!

We will soon see that there exist cryptosystems which are believed to be secure. While none of these beliefs are proven, we do know that certain of these are in fact secure (if implemented correctly) if and only if a certain important mathematical problem cannot be easily solved.

- So, to crack such a system, one has to solve a mathematical problem that many people care about deeply. If this happens, you will most likely read about it in the (academic) news, and you will have an opportunity to update your system in time (most likely, you'll hear about progress much earlier).
- On the other hand, if you use a cryptosystem that is not well-studied, then it may well happen that an adversary breaks your system and keeps exploiting the security leak without you ever learning about it.

Not particularly related but important to keep in mind:

Frequently, security's weakest link are humans. It's very hard to protect against that.
https://en.wikipedia.org/wiki/Social_engineering_(security)

Review: Chinese remainder theorem

Example 64. (warmup)

(a) If $x \equiv 3 \pmod{10}$, what can we say about $x \pmod 5$?

(b) If $x \equiv 3 \pmod 7$, what can we say about $x \pmod 5$?

Solution.

(a) If $x \equiv 3 \pmod{10}$, then $x \equiv 3 \pmod 5$.
[Why?! Because $x \equiv 3 \pmod{10}$ if and only if $x = 3 + 10m$, which modulo 5 reduces to $x \equiv 3 \pmod 5$.]

(b) Absolutely nothing! $x = 3 + 7m$ can be anything modulo 5 (because $7 \equiv 2$ is invertible modulo 5).

Example 65. If $x \equiv 32 \pmod{35}$, then $x \equiv 2 \pmod 5$, $x \equiv 4 \pmod 7$.

Why?! As in the first part of the warmup, if $x \equiv 32 \pmod{35}$, then $x \equiv 32 \pmod 5$ and $x \equiv 32 \pmod 7$.

The Chinese remainder theorem says that this can be reversed!

That is, if $x \equiv 2 \pmod 5$ and $x \equiv 4 \pmod 7$, then the value of x modulo $5 \cdot 7 = 35$ is determined.
[How to find the exact value of x, namely $x \equiv 32 \pmod{35}$, is discussed in the next example.]

Example 66. Solve $x \equiv 2 \pmod 5$, $x \equiv 4 \pmod 7$.

Solution. $x \equiv 2 \cdot 7 \cdot \underbrace{7^{-1}_{\bmod 5}}_{3} + 4 \cdot 5 \cdot \underbrace{5^{-1}_{\bmod 7}}_{3} \equiv 42 + 60 \equiv 32 \pmod{35}$

Important. Can you see how we need 5 and 7 to be coprime here?

Brute-force solution. Note that, while in principle we can always perform a brute-force search, this is not practical for larger problems. Here, if x is a solution, then so is $x + 35$. So we only look for solutions modulo 35.
Since $x \equiv 4 \pmod 7$, the only candidates for solutions are $4, 11, 18, \ldots$ Among these, we find $x = 32$.
[We can also focus on $x \equiv 2 \pmod 5$ and consider the candidates $2, 7, 12, \ldots$, but that is even more work.]

Example 67. Solve $x \equiv 1 \pmod 4$, $x \equiv 2 \pmod 5$.

Solution. $x \equiv 1 \cdot 5 \cdot \underbrace{5^{-1}_{\bmod 4}}_{1} + 2 \cdot 4 \cdot \underbrace{4^{-1}_{\bmod 5}}_{-1} \equiv 5 - 8 \equiv -3 \pmod{20}$

Example 68. Solve $x \equiv 1 \pmod 4$, $x \equiv 2 \pmod 5$, $x \equiv 3 \pmod 7$.

Solution. (option 1) By the previous problem, the first two congruences combine to $x \equiv -3 \pmod{20}$.
Using $x \equiv -3 \pmod{20}$, $x \equiv 3 \pmod 7$, we find $x \equiv -3 \cdot 7 \cdot \underbrace{7^{-1}_{\bmod 20}}_{3} + 3 \cdot 20 \cdot \underbrace{20^{-1}_{\bmod 7}}_{-1} \equiv -63 - 60 \equiv 17 \pmod{140}$.

Solution. (option 2) $x \equiv 1 \cdot 5 \cdot 7 \cdot \underbrace{[(5 \cdot 7)^{-1}_{\bmod 4}]}_{-1} + 2 \cdot 4 \cdot 7 \cdot \underbrace{[(4 \cdot 7)^{-1}_{\bmod 5}]}_{2} + 3 \cdot 4 \cdot 5 \cdot \underbrace{[(4 \cdot 5)^{-1}_{\bmod 7}]}_{-1} \equiv 17 \pmod{140}$

Theorem 69. (Chinese Remainder Theorem) Let $n_1, n_2, \ldots, n_r$ be positive integers with $\gcd(n_i, n_j) = 1$ for $i \neq j$. Then the system of congruences

$$x \equiv a_1 \pmod{n_1}, \quad \ldots, \quad x \equiv a_r \pmod{n_r}$$

has a simultaneous solution, which is unique modulo $n = n_1 \cdots n_r$.

In other words. The Chinese remainder theorem provides a bijective (i.e., 1-1 and onto) correspondence

$$x \pmod{nm} \mapsto \begin{bmatrix} x \pmod n \\ x \pmod m \end{bmatrix}$$

provided that m and n are coprime.

For instance. Let's make the correspondence explicit for $n = 2$, $m = 3$:
$0 \mapsto \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $1 \mapsto \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $2 \mapsto \begin{bmatrix} 0 \\ 2 \end{bmatrix}$, $3 \mapsto \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $4 \mapsto \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $5 \mapsto \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

Armin Straub
straub@southalabama.edu