Alternative Approaches: Bounded Storage Model

A. Würfl

17th April 2005

Motivation

Common practice in cryptography:

If you need an encryption scheme then just take AES (or RSA, ElGamal, . . .) and you can be confident that your adversary cannot break it.

Why can you be confident?

Because several people tried to break these ciphers and nobody succeeded.

Question:

Can we also be confident that nobody will break it in the future?

Will the current ciphers be secure in the future?

There is a good chance they will not!

One of the following disasters can happen:

- computing power will increase dramatically
- some non-standard computation model will be implemented (e.g. quantum computers)
- certain computational tasks (e.g. factoring) will turn out to be easier than expected
  (or simply someone will show $P = NP$)

Possible Attack

So the following attack is possible:

1. The adversary stores the transcript of your entire communication
2. Later (maybe in 30 years) he decrypts it.

Historical Example: VENONA Project

Example

In 1942-46 Americans read and stored a large number of Soviet cryptograms.
Some of them were decrypted decades later.

Main Idea

Instead of assuming that the computing power of the adversary is limited we assume that the memory is limited.

Strategy of the Strongly-Randomized Cipher

The Bounded-Storage-Model is based on a strongly-randomized cipher.
It was proposed by Maurer in 1992

- new concept of proveable security
- use of publicly-accesible string of random bits
- artificial blow-up of data

Terminology

Definition

- capital letters denote random variables, underlined denote random vectors
- $\underline{R}$ is the publicly accessible random string of bits
- $\underline{X}$ denotes the plaintext, $\underline{Y}$ the cryptogram and $\underline{W}$ the keystream
- $\underline{X}$, $\underline{Y}$ and $\underline{W}$ are of length N

Preparation

- Take $L = KT$ bits of $\underline{R}$ and form a two-dimensional array.
- Denote the entries with:

$$\begin{array}{|cccc|} R[1,0] & R[1,1] & \dots & R[1,T-1] \\ R[2,0] & R[2,1] & \dots & R[2,T-1] \\ \vdots & \vdots & & \vdots \\ R[K,0] & R[K,1] & \dots & R[K,T-1] \end{array}$$

The secret Key

The secret Key

$\underline{Z} = [Z_1, \ldots, Z_K]$, where $Z_k \in \{0, \ldots, T-1\}$ for $1 \leq k \leq K$

- specifies a position within each row of $\underline{R}$
- uniformly distributed over the key space ($S_{\underline{Z}} = \{0, \ldots, T-1\}^K$)
- key-length: $K \cdot \log_2 T$

The Keystream $\underline{W}$

Key-function

Generating the keystream $\underline{W}$ from the secret key $\underline{Z}$ and the randomizer $\underline{R}$.

Definition

$$W_n = \left(\sum_{k=1}^{K} R[k, (n - 1 + Z_k) \mod T] \right) \mod 2$$

where each row of $\underline{R}$ is considered to be extended cyclically

And finally. . .

The plaintext $\underline{X}$ and the keystream $\underline{W}$ are added bitwise modulo 2:

Definition

$$Y_n = X_n \oplus W_n \text{ for } 1 \leq n \leq N$$

$$\oplus : a \oplus b := (a + b) \mod 2$$

Example

Example

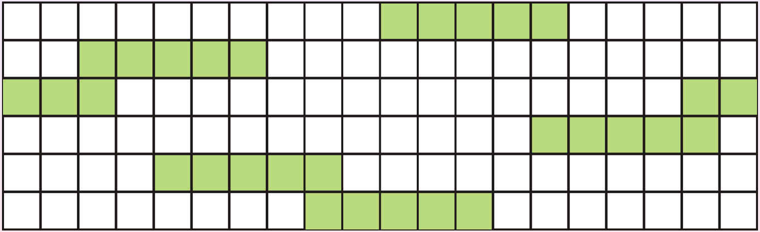
Let $T = 20, K = 6$, $N = 5$, $Z = [11, 3, 19, 15, 5, 9]$

Resulting keystream:

Example

Example

Let $T = 20, K = 6$, $N = 5$, $Z = [11, 3, 19, 15, 5, 9]$



Resulting keystream:

Example

Example

Let $T = 20, K = 6, N = 5, Z = [11, 3, 19, 15, 5, 9]$

										0	1	0	1	1					
		1	1	0	1	1													
0	0	1																1	0
														0	1	0	0	1	
				0	1	0	1	0											
								1	1	1	0	0							

Resulting keystream:

Example

Example

Let $T = 20, K = 6, N = 5, Z = [11, 3, 19, 15, 5, 9]$

										0	1	0	1	1					
		1	1	0	1	1													
0	0	1																1	0
														0	1	0	0	1	
				0	1	0	1	0											
								1	1	1	0	0							

Resulting keystream:

1				

Example

Example

Let $T = 20, K = 6, N = 5, Z = [11, 3, 19, 15, 5, 9]$

										0	1	0	1	1					
		1	1	0	1	1													
0	0	1																1	0
														0	1	0	0	1	
				0	1	0	1	0											
								1	1	1	0	0							

Resulting keystream:

1	1			

Example

Example

Let $T = 20, K = 6, N = 5, Z = [11, 3, 19, 15, 5, 9]$

										0	1	0	1	1					
		1	1	0	1	1													
0	0	1																1	0
														0	1	0	0	1	
				0	1	0	1	0											
								1	1	1	0	0							

Resulting keystream:

1	1	1		

Example

Example

Let $T = 20, K = 6, N = 5, Z = [11, 3, 19, 15, 5, 9]$

										0	1	0	1	1					
		1	1	0	1	1													
0	0	1																1	0
													0	1	0	0	1		
				0	1	0	1	0											
								1	1	1	0	0							

Resulting keystream:

1	1	1	1	

Example

Example

Let $T = 20, K = 6, N = 5, Z = [11, 3, 19, 15, 5, 9]$

										0	1	0	1	1					
		1	1	0	1	1													
0	0	1																1	0
														0	1	0	0	1	
				0	1	0	1	0											
								1	1	1	0	0							

Resulting keystream:

1	1	1	1	0

Main Result by Maurer '92

Theorem

Only with probability of $1 - N\delta^K$ an eavesdropper will obtain any information about the plaintext where $\delta = M/KT$

M: Number of bits examined by the eavesdropper

Very simplistic! See paper for exact theorem and proof.

This means:

An example with practical relevance:

- $K = 50$, $T = 2^{50} \approx 10^{15}$
  plaintext: 1 Gbit, i.e.: $N = 2^{30} \approx 10^9$.
- **resulting keysize**: $50 \cdot \log_2 2^{50} = 2500$ bits
- legitimate users need to examine only 50 randomizer bits per plaintext bit
- eavesdropper examines $\delta = 1/2$ of all bits, i.e., $M = KT/2 = 25 \cdot 2^{50}$ bits
- chance of obtaining any new information about the plaintext: not greater than $2^{30} \cdot (1/2)^{50} < 10^{-6}$

Application

- Generating a random string using a deep-space radio-source
- Satellite broadcasting random bit-stream with high bandwidth (not yet implemented)
- Current limit for adversary's memory: 2^{50} Byte cost 1 Billion $
- No possibility to decode cryptogram later

Thank you for your Attention!