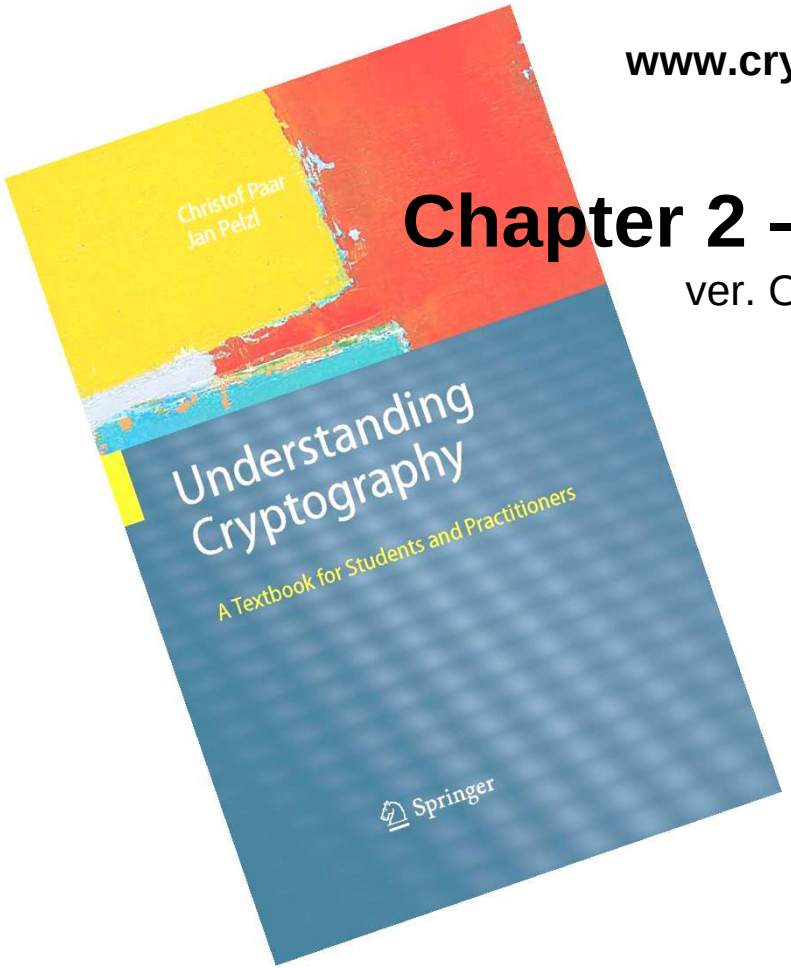
Bilgisayar Güvenliği ve Kriptografi BLM5101 - Gr 1		
Hafta	Tarih	Konular
1	17.Şub	Giriş
2	24.Şub	Kriptografi-Bilgi Güvenliği Temeller
3	03.Mar	Akan Şifreleme
4	10.Mar	DES – AES - **Paper seçim**
5	17.Mar	Blok Şifreleme - Public key kripto
6	24.Mar	RSA
7	31.Mar	Diskrit Logaritma Tabanlı Public-Key Kriptografi
8	07.Nis	Eliptic Curve Kriptografi
9	**14.Nis**	**1. Ara Sınav**
10	21.Nis	Dijital İmza
11	28.Nis	Hash Fonksiyonlar
12	05.May	SUNUM
13	12.May	SUNUM
14	**19.May**	**Tatil**
15	26.May	SUNUM

Understanding Cryptography – A Textbook for Students and Practitioners

by Christof Paar and Jan Pelzl

www.crypto-textbook.com



Chapter 2 – Stream Ciphers

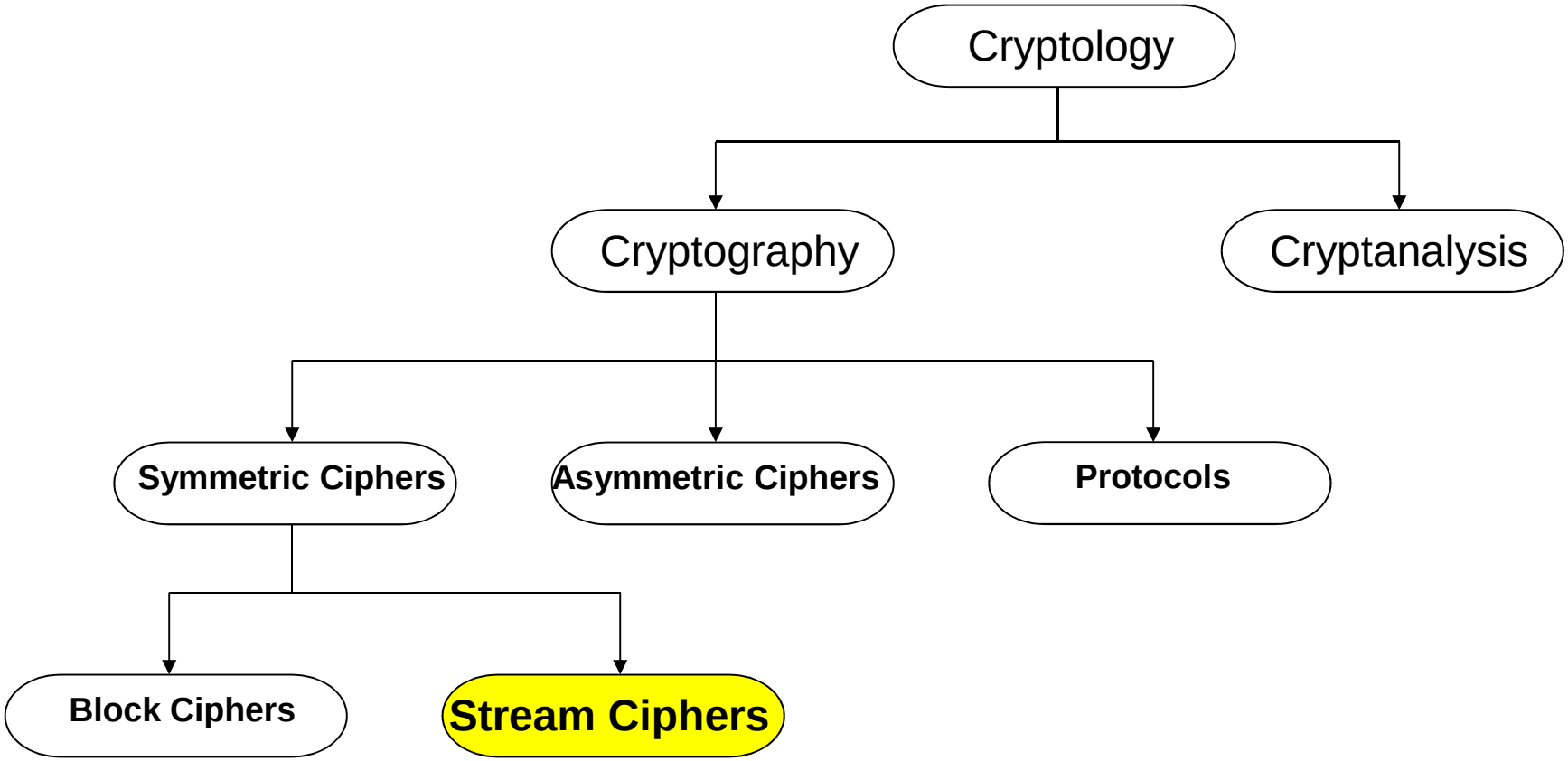
ver. October 29, 2009

These slides were prepared by Thomas Eisenbarth, Christof Paar and Jan Pelzl

Content of this Chapter

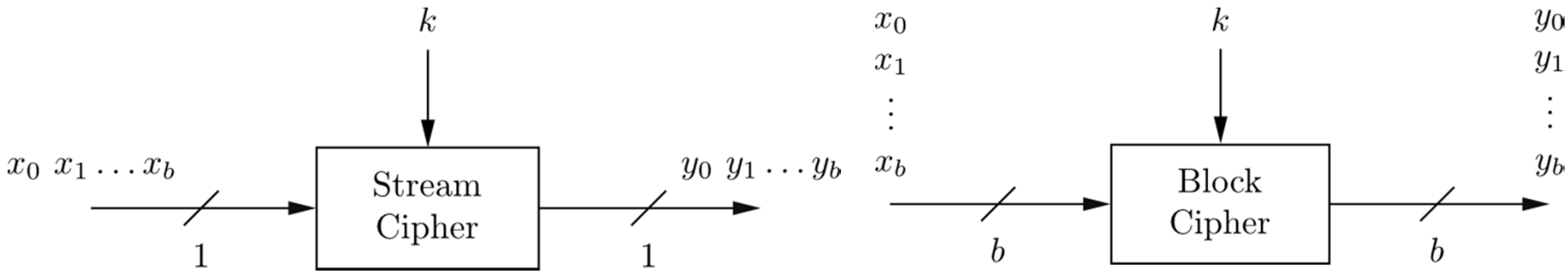
- **Intro to stream ciphers**
- Random number generators (RNGs)
- One-Time Pad (OTP)
- Linear feedback shift registers (LFSRs)
- Trivium: a modern stream cipher

Stream Ciphers in the Field of Cryptology



Stream Ciphers were invented in 1917 by Gilbert Vernam

Stream Cipher vs. Block Cipher



- **Stream Ciphers- Akan Şifreleme**
  - Encrypt bits individually – bitleri teker teker şifreler
  - Usually small and fast → common in embedded devices (e.g., A5/1 for GSM phones)
- **Block Ciphers:**
  - Always encrypt a full block (several bits)
  - Her seferinde açık metindeki bütün bit bloğunu aynı anahtarla şifreler. Dolayısı ile herhangi bir açık metin bitinin şifrelenmesi aynı bloktaki diğer bitlere bağlıdır.
  - Pratikte blok şifreleyicilerin çoğunda blok uzunluğu 128 (AES) veya 64 bit (DES)

Stream Cipher vs. Block Cipher

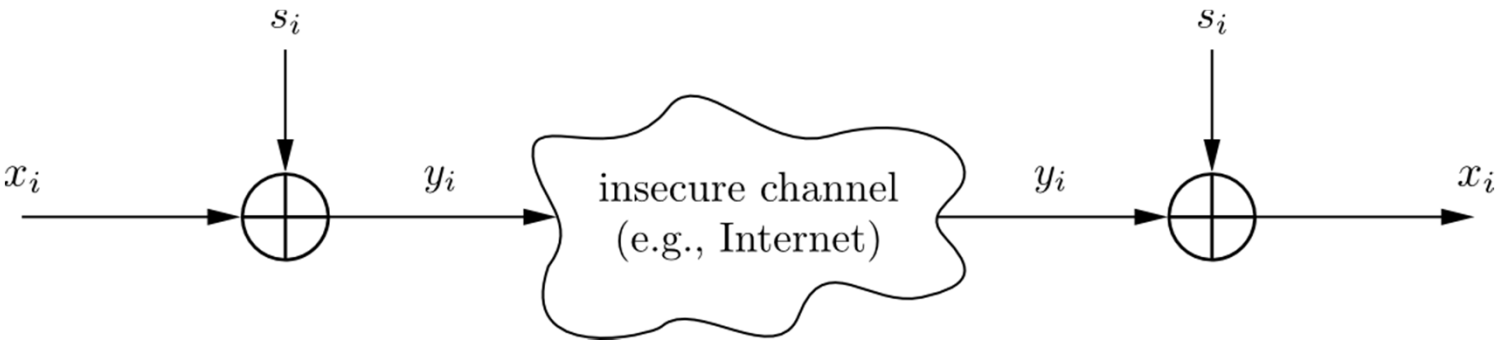
- Geleneksel olarak akan şifreleme sistemlerinin blok şifreleme sistemlerinden daha ‘etkin’ olduğu varsayılır:
- Yazılımsal olarak optimize edilmiş akan şifreleme sistemleri, bir biti şifrelemek için, daha az prosesör komutu (daha az cycle) kullanır
- Donanımsal olarak optimize edilmiş akan şifreleme sistemleri blok şifrelemeye göre, aynı oranda veriyi şifrelemek için, daha az kapı (gate) veya daha küçük chip alanı gerektirir.

- Ancak AES gibi modern blok şifreleme yöntemleri yazılımsal olarak çok etkindir. Benzer şekilde donanımsal olarak çok etkin blok şifreleyiciler de mevcuttur; PRESENT gibi...

Encryption and Decryption with Stream Ciphers

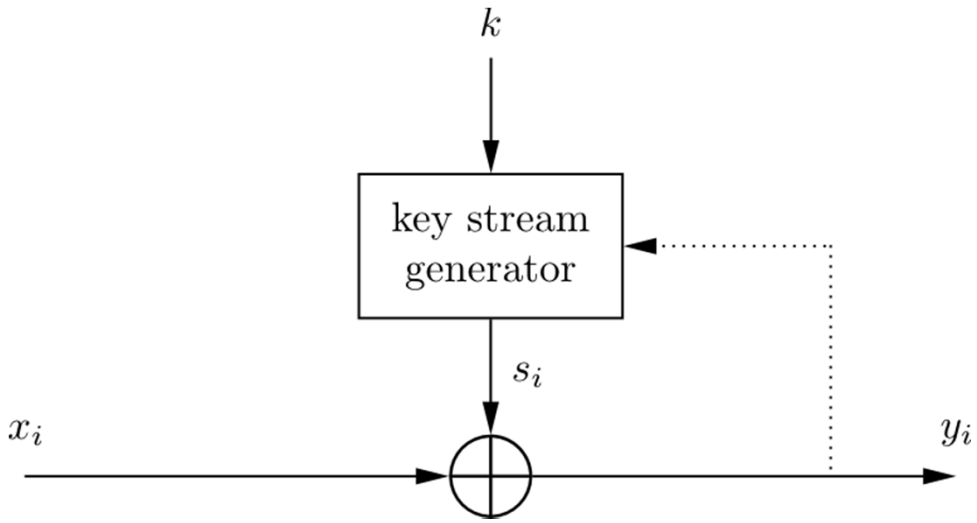
Plaintext x_i, ciphertext y_i and key stream s_i consist of individual bits

Şifresiz metin, şifreli metin ve akan anahtar ayrı ayrı bitlerden oluşuyor



- Encryption and decryption are simple additions modulo 2 (aka XOR)
- Encryption and decryption are the same functions
- **Encryption:** $y_i = e_{si}(x_i) = x_i + s_i \bmod 2$ $\qquad x_i, y_i, s_i \in \{0,1\}$
- **Decryption:** $x_i = e_{si}(y_i) = y_i + s_i \bmod 2$
- Akan anahtardan gelen bitin (s_i) 2'ye göre modu alınarak o anda şifrelenecek açık metine ait bite (x_i) ekleniyor
- Deşifreleme de aynı fonksiyon kullanılarak yapılıyor
- $\oplus \longrightarrow$ XOR veya 2'ye göre mod alarak toplama işlemi

Synchronous vs. Asynchronous Stream Cipher



- Security of stream cipher depends entirely on the key stream s_i :
  - Should be **random** , i.e., $Pr(s_i = 0) = Pr(s_i = 1) = 0.5$ – anahtardaki her bitin 0 veya 1 değeri alma olasılı eşit
  - Must be **reproducible** by sender and receiver
- **Synchronous Stream Cipher**
  - Key stream depend only on the key (and possibly an initialization vector IV)
- **Asynchronous Stream Ciphers**
  - Key stream depends also on the ciphertext (dotted feedback enabled)

Why is Modulo 2 Addition a Good Encryption Function?

- Modulo 2 addition is equivalent to XOR operation

- For perfectly random key stream s_i , each ciphertext output bit has a 50% chance to be 0 or 1

  → Good statistic property for ciphertext

x_i	s_i	y_i
0	0	0
0	1	1
1	0	1
1	1	0

Tablonun ilk satırında şifrelenecek bit $x_i = 0$ ve şifrelenmiş bit $y_i = 0$ çıkıyor
İkinci satırda şifrelenecek bit yine $x_i = 0$ ama şifrelenmiş bit $y_i = 1$ çıkıyor.
Aynı veriyi şifrelerken sonucun 0 veya 1 çıkma olasılığı aynı

Inverting XOR is simple, since it is the same XOR operation

ÖRNEK

- Alice ‘A’ harfini akan şifreleme yöntemi ile şifrelemek istiyor – harf ASCII kod olarak verilmiş:
- A Ascii 65
- $65_{10} = 1000001_2$
- Anahtar akışı $\longrightarrow$ $(s_0, s_1, \ldots\ldots s_6)= 0101100$ olsun

$x_{0,\ldots,} x_{6=} 1000001$

$\oplus$

$s_{0,\ldots,} s_{6=} 1000001$

} Alice

1101101 = m (109 Ascii) $\longrightarrow$ Oscar’ın göreceği

$y_{0,\ldots,} y_{6=} 1101101$

$\oplus$

$s_{0,\ldots,} s_{6=} 1000001$

$x_{0,\ldots,} x_{6=} 1000001 = A$

} Bob

- Akan şifreleme sistemlerinde esas konu akan anahtarın güvenliğidir
- Akan anahtarın (key stream) nasıl üretildiği akan şifreleme sisteminin güvenilirliğini belirler
- En önemli gereklilik key stream'deki bitlerin saldırgana rasgele bir dizi gibi görinmesidir
- Key stream tahmin edilebilirse şifre çözülebilir

- Akan şifreleme 1917'de Gilbert Verman tarafından bulunmuş, o zaman bir adı yokmuş, sonradan Verman Şifreleyici de deniyor

- Verman yaptığı elektromekanik makineye şifresiz metin ve akan anahtarı bantlar halinde veriyor. İlk kez şifreleme ve gönderme (transmission) tek makinede otomatik hale getirilmiş.

Stream Cipher: Throughput

Performance comparison of symmetric ciphers (Pentium4):

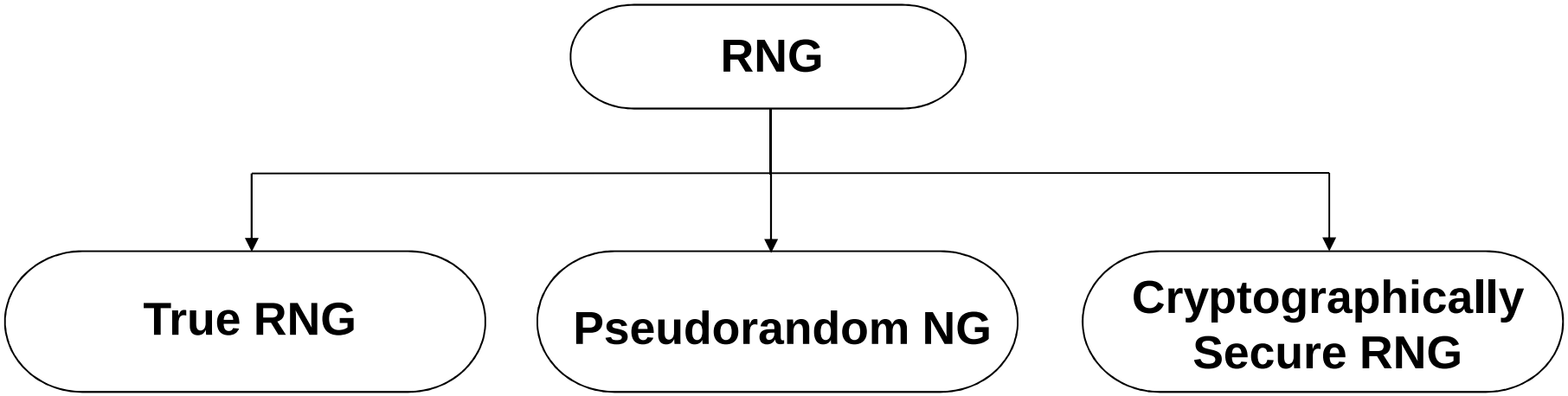
Cipher	Key length	Mbit/s
DES	56	36.95
3DES	112	13.32
AES	128	51.19
RC4 (stream cipher)	(choosable)	211.34

Source: Zhao et al., Anatomy and Performance of SSL Processing, ISPASS 2005

Content of this Chapter

- Intro to stream ciphers
- **Random number generators (RNGs)**
- One-Time Pad (OTP)
- Linear feedback shift registers (LFSRs)
- Trivium: a modern stream cipher

Random number generators (RNGs)



True Random Number Generators (TRNGs) – Gerçek Rasgele Sayı Üreteçleri

- Based on physical random processes:
- coin flipping – para atma
- dice rolling – zar atma
- semiconductor noise –yarı iletken gürültüsü
- radioactive decay – radyoaktif bozunma
- mouse movement – mouse hareketi
- clock jitter of digital circuits – sayısal devrelerde clock titreşimi

- Output stream s_i should have good statistical properties:
  $\Pr(s_i = 0) = \Pr(s_i = 1) = 50\%$ (often achieved by post-processing)

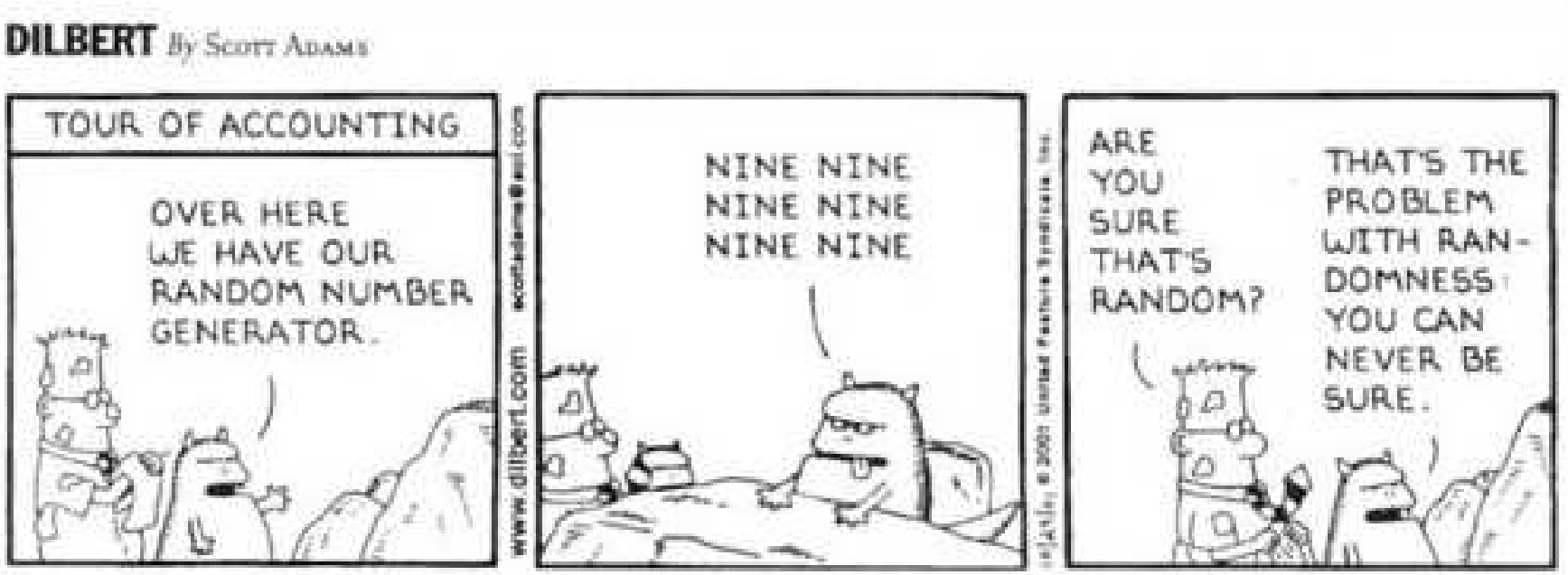
Output can neither be predicted nor be reproduced

True Random Number Generators (TRNGs) - Gerçek Rasgele Sayı Üreteçleri

- Gerçek rasgele sayı üreteçlerinin temel özelliği, çıktıların tekrar üretilebilir olmamasıdır.
- Örneğin 100 kez para atıp sonuçları kaydettiğimizde gerçek bir rasgele sayı dizisi oluşturmuş oluruz. Teorik olarak dünyada hiç kimsenin aynı 100 bitlik seriyi üretmesi imkansız – esasen bu ihtimal $\frac{1}{2^{100}}$ yani çokçok küçük !!

- Typically used for generation of keys, nonces (number used once or used only once values) and for many other purposes
- Genellikle dönemsel tek kullanımlık anahtar üretme için kullanılır ve alıcı ile göndericiye dağıtılır.

Gerçek Rasgele Sayı Üreteçleri



Pseudorandom Number Generator (PRNG) - Sözde Rasgele Sayı Üreteçleri

- Generate sequences from initial seed value – bi kök değerinden hesaplama yolu ile sayı dizisi üretiyor
- Typically, output stream has good statistical properties – çıktıların istatiksel özellikleri iyi
- Output can be reproduced and can be predicted – ancak çıktılar yeniden üretilebilir ve tahmin edilebilir
- Often computed in a recursive way – genellikle özyinemeli yöntemler:

$$s_0 = seed$$

$$s_{i+1} = f(s_i, s_{i-1}, \ldots, s_{i-t})$$

Example: *rand() function* in ANSI C:

$$s_0 = 12345$$

$$s_{i+1} = 1103515245 s_i + 12345 \bmod 2^{31}$$

Most PRNGs have bad cryptographic properties!

Cryptanalyzing a Simple PRNG

Simple PRNG: **Linear Congruential Generator**

$$S_0 = seed$$

$$S_{i+1} = AS_i + B \bmod m$$

A – çarpan
B – artım
m – modül
s_0, A ve B değerlerinin seçimi çevrim uzunluğunu (periyodu) önemli derecede etkiler

Assume

- unknown *A*, *B and* S_0 as key
- Size of *A*, *B* and S_i to be 100 bit
- 300 bit of output are known, i.e. S_1, S_2 and S_3 - **elimizde anahtara ait 3 bit var**

Solving

$$S_2 = AS_1 + B \bmod m$$
$$S_3 = AS_2 + B \bmod m$$

Lineer denklem seti

2 bilinmeyen 2 denklem

…directly reveals A and B. All S_i can be computed easily!

Bad cryptographic properties due to the linearity of most PRNGs

Cryptanalyzing a Simple PRNG

- PRNG - sözde rasgele sayı üreteçleri gerçek manada ‘rasgele değil ‘, hesaplanabiliyor dolaysı ile ‘deterministik’

- PRNG’lerin iyi istatistik özelliklere sahip olması beklenir: uniformluk ve bağımsızlık özellikleri olmalı, çıktıları gerçek rasgele sayı dizilerine benzemeli
- Chi-Square gibi matematiksel testlerle istatistiksel özellikleri doğrulanabilir.
- Kriptografi dışında pek çok uygulama alanı var; simulasyonlar, test yazılımları vb..

Cryptographically Secure Pseudorandom Number Generator (CSPRNG) - Kriptografik Olarak Güvenli Sözde Rasgele Sayı Üreteçleri

- Special PRNG with additional property:
  - Output must be **unpredictable**

More precisely: Given n consecutive bits of output s_i , the following output bits s_{n+1} cannot be predicted (in polynomial time).

- Needed in cryptography, in particular for stream ciphers

- Remark: There are almost no other applications that need unpredictability, whereas many, many (technical) systems need PRNGs.

Kriptografik Olarak Güvenli Sözde Rasgele Sayı Üreteçleri

- PRNG'a ek olarak ' tahmin edilemez' olma özelliği taşımalı:
- Anahtar akışına ait $s_i, s_{i+1}, ..., s_{i+n-1}$ gibi n bit verildiğinde sonraki bitlerin tahmin edilmesi hesapsal olarak mümkün olmamalı
- Diğer bir deyişle; ardışıl n biti verilen anahtar akışına ait takip eden biti (s_{n+1})

%50 den daha fazla başarı ihtimali ile hesaplayabilecek polinomsal zamanlı (Adım sayısının bir polinom ile ifade edilebildiği algoritmalar) bir algoritma yoktur.

Polinomsal zaman
- Sabit zaman
- Lineer zaman
- ikinci derece zaman
- .....

Logaritmik zamanlı / üstel zamanlı / NP-complete algoritmalar uygun...

'Tahmin Edilememe' ihtiyacı kriptografiye özeldir. Diğer bilimsel ve mühendislik uygulamalarında böyle bir ihtiyaç yok, dolayısı ile PRNG ve CSPRNG arasındaki fark kriptografi ile uğraşmayanlar tarafından pek bilinmiyor !!

Content of this Chapter

- Intro to stream ciphers
- Random number generators (RNGs)
- **One-Time Pad (OTP)**
- Linear feedback shift registers (LFSRs)
- Trivium: a modern stream cipher

One-Time Pad (OTP)

Unconditionally secure cryptosystem:

- A cryptosystem is unconditionally secure if it cannot be broken even with *infinite* computational resources

One-Time Pad

- A cryptosystem developed by Mauborgne that is based on Vernam's stream cipher:
- Properties:

  Let the plaintext, ciphertext and key consist of individual bits $x_i, y_i, k_i \in \{0,1\}$.

Encryption:	$e_{k_i}(x_i) = x_i \oplus k_i$.
Decryption:	$d_{k_i}(y_i) = y_i \oplus k_i$

OTP is unconditionally secure if and only if the key k_i. is used once!

Mutlak Güvenlik – Unconditional Security

- Mutlak güvenlik, bilgi teorisine dayanır ve saldırganın sınırsız hesaplama gücüne sahip olduğunu varsayar.
- Bu çok net bir tanım olmakla birlikte gereklilikleri oldukça fazla;
- Örneğin bir simetrik şifreleme algoritmamız olsun (blok veya akan şifre olması farketmez)
- Anahtar uzunluğu 10.000 bit olsun ve tek olası saldırı yöntemi kaba kuvvet saldırısı olsun
- Bu durumda saldırganın sonsuz hesapsal kaynağa sahip olabileceğini – saldırganın 2^{10000} bilgisayarı olabileceğini ve her bilgisayarın bir olası anahtarı kontrol edebileceğini varsaymak zorundayız. Yani saldırgan 1 adımda doğru anahtarı bulabilir.
- Elbette 2^{10000} bilgisayarın üretilmesi imkansız – evrende bilindiği kadarı ile 2^{266} civarında atom var !!
- Yine de bu tür bir şifrelemeye ancak **hesapsal olarak güvenli** diyebiliriz ama **mutlak güvenli** diyemeyiz

One Time Pad – Tek kerelik blok

- Bu tanımlamalar bize mutlak güvenli bir şifreleyicinin nasıl yapılabileceğinin yolunu gösteriyor:

1- Anahtar akışı (s_0, s_1,s_2,.....) gerçek rasgele sayı üreteci (TRNG) ile üretilir

2- Anhtar akışı sadece resmi olarak haberleşen taraflarca bilinir

3- Anahtar akışındaki her bit (s_i) sadece bir kere kullanılır.

Bu tek kerelik blok (OTP) olarak adlandırılır ve mutlak güvenlidir.

One-Time Pad (OTP)

Unconditionally secure cryptosystem:

$$y_0 = x_0 \oplus s_0 \bmod 2 = x_0 \oplus k_0$$
$$y_1 = x_1 \oplus s_1 \bmod 2 = x_1 \oplus k_1$$
$$\ldots\ldots\ldots\ldots$$

Bu denklemlerin her biri 2 bilinmeyenli lineer denklemler ve birbirinden bağımsız

Every equation is a linear equation with two unknowns

$\Rightarrow$ for every y_i are $x_i = 0$ and $x_i = 1$ equiprobable(equally probable)

Saldırgan y_0 değerini (0 veya 1) bilse de x_0 değerini ancak %50 ihtimalle bilebilir.

$\Rightarrow$This is true if k_0, k_1, ... are independent, i.e., all k_i have to be generated truly random

s_i değerleri dolayısı ile k_i değerleri ' gerçek rasgele' olmasaydı durum farklı olurdu - s_i değerleri arasında fonksiyonel bir ilişki olurdu ve yukarıdaki denklemler birbirinden bağımsız olmazdı.

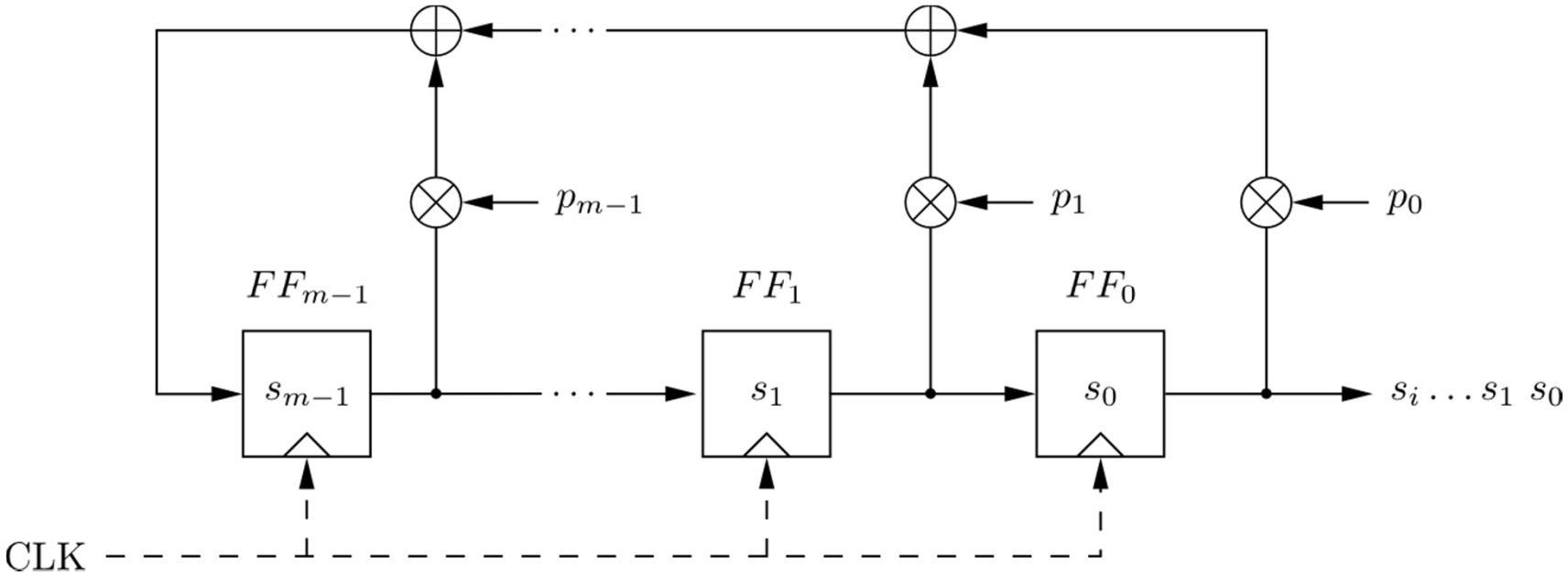
$\Rightarrow$ It can be shown that this systems can *provably* not be solved.

Disadvantage: For almost all applications the OTP is **impractical** since the key must be as long as the message! (Imagine you have to encrypt a 1GByte email attachment.)

Content of this Chapter

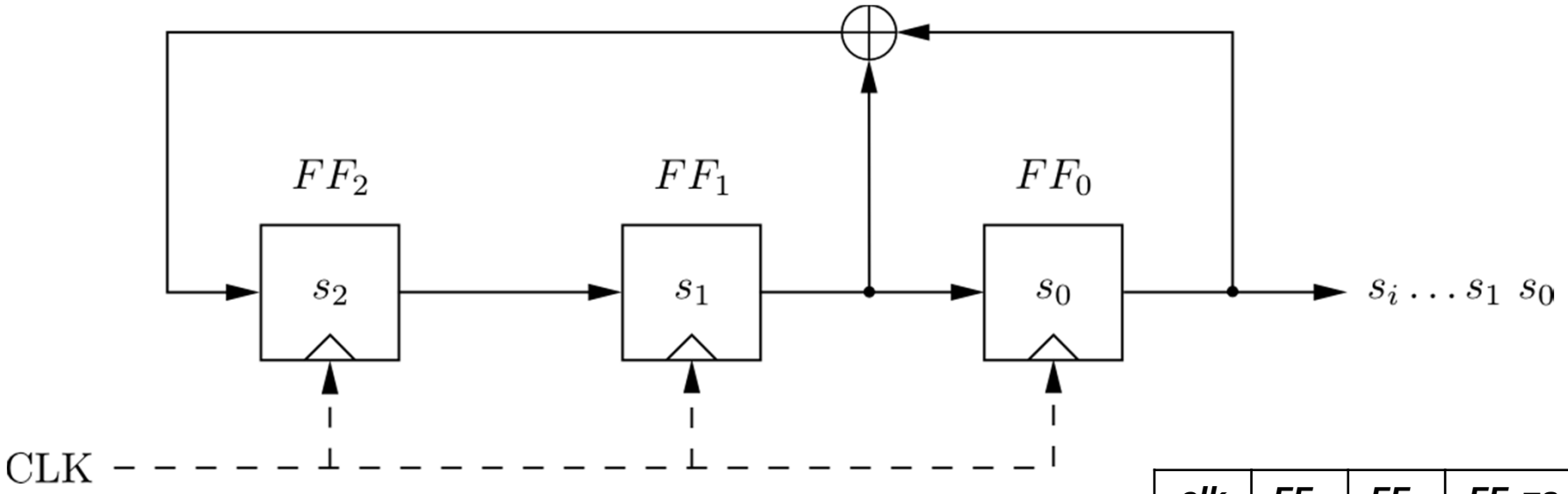
- Intro to stream ciphers
- Random number generators (RNGs)
- One-Time Pad (OTP)
- **Linear feedback shift registers (LFSRs)**
- Trivium: a modern stream cipher

Linear Feedback Shift Registers (LFSRs)



- Concatenated *flip-flops (FF*), i.e., a shift register together with a feedback path
- Feedback computes fresh input by XOR of certain state bits
- *Degree m* given by number of storage elements
- If $p_i = 1$, the feedback connection is present ("closed switch), otherwise there is not feedback from this flip-flop ("open switch")
- Output sequence repeats periodically
- Maximum output length**:** 2^m-1

Linear Feedback Shift Registers (LFSRs): Example with m=3



clk	FF_2	FF_1	$FF_0=s_i$
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

- LFSR output described by recursive equation:

$$s_{i+3} = s_{i+1} + s_i \bmod 2$$

- Maximum output length (of 2^3-1=7) achieved only for certain feedback configurations, .e.g., the one shown here.

Security of LFSRs

LFSRs typically described by polynomials:

$$P(x) = x^m + p_{l-1}x^{m-1} + \ldots + p_1 x + p_0$$

- Single LFSRs generate highly predictable output
- If 2*m* output bits of an LFSR of degree *m* are known, the feedback coefficients p_i of the LFSR can be found by solving a system of linear equations*

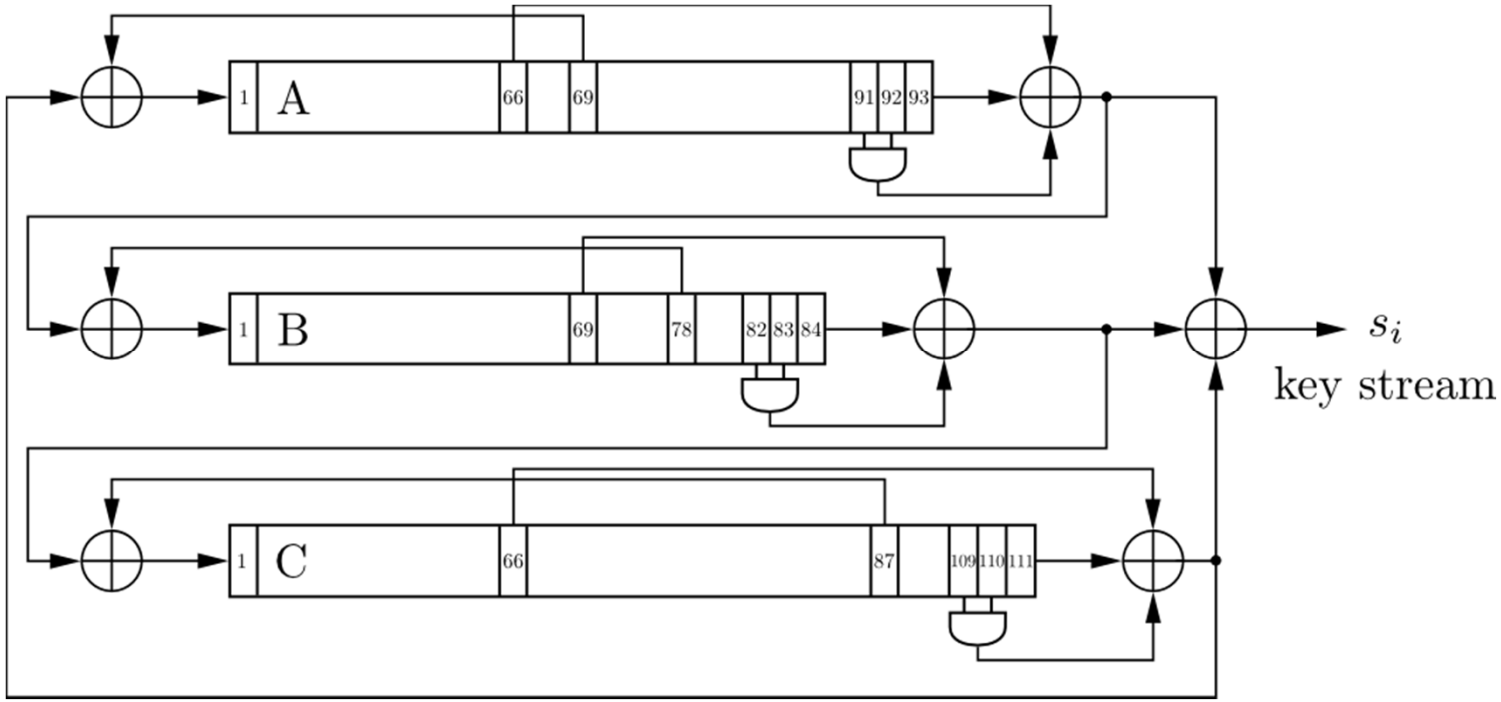
- Because of this many stream ciphers use **combinations** of LFSRs

*See Chapter 2 of *Understanding Cryptography* for further details.

Content of this Chapter

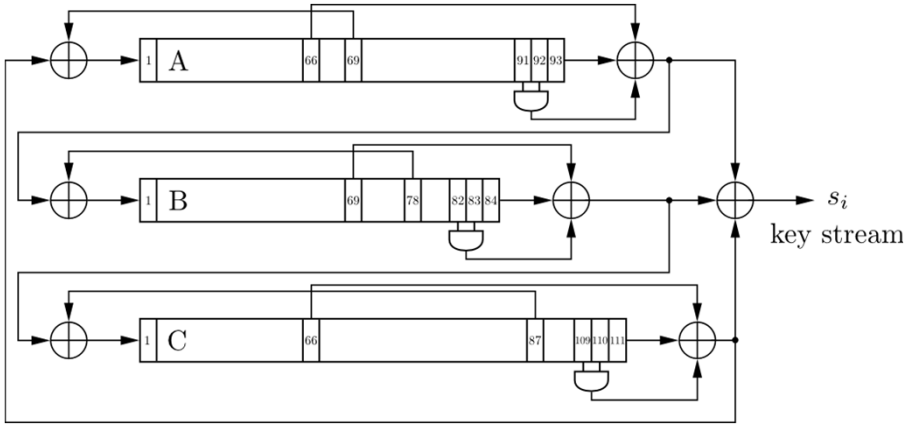
- Intro to stream ciphers
- Random number generators (RNGs)
- One-Time Pad (OTP)
- Linear feedback shift registers (LFSRs)
- **Trivium: a modern stream cipher**

A Modern Stream Cipher - Trivium



- Three *nonlinear* LFSRs (NLFSR) of length 93, 84, 111
- XOR-Sum of all three NLFSR outputs generates key stream s_i
- Small in Hardware:
  - Total register count: 288
  - Non-linearity: 3 AND-Gates
  - 7 XOR-Gates (4 with three inputs)

Trivium



Initialization:

- Load 80-bit IV into A
- Load 80-bit key into B
- Set c_{109} , c_{110} , c_{111} =1, all other bits 0

Warm-Up:

- Clock cipher 4 x 288 = 1152 times without generating output

Encryption:

- XOR-Sum of all three NLFSR outputs generates key stream s_i

Design can be parallelized to produce up to 64 bits of output per clock cycle

	Register length	Feedback bit	Feedforward bit	AND inputs
A	93	69	66	91, 92
B	84	78	69	82, 83
C	111	87	66	109, 110

Lessons Learned

- Stream ciphers are less popular than block ciphers in most domains such as Internet security. There are exceptions, for instance, the popular stream cipher RC4.

- Stream ciphers sometimes require fewer resources, e.g., code size or chip area, for implementation than block ciphers, and they are attractive for use in constrained environments such as cell phones.

- The requirements for a *cryptographically secure pseudorandom number generator* are far more demanding than the requirements for pseudorandom number generators used in other applications such as testing or simulation

- The One-Time Pad is a provable secure symmetric cipher. However, it is highly impractical for most applications because the key length has to equal the message length.

- Single LFSRs make poor stream ciphers despite their good statistical properties. However, careful combinations of several LFSR can yield strong ciphers.