Lecture 9: Pseudorandom Numbers and Stream Ciphers

COSC362 Data and Network Security

Book 1: Chapter 8 – Book 2: Chapter 20

Spring Semester, 2021

Motivation

- Random values required in many cases in cryptography.
- For practical reasons, pseudorandom deterministic algorithms are used.
- Stream ciphers constructed from (pseudo)random number generators.
- Examples of stream ciphers widely deployed:
  - A5 cipher used in GSM mobile phones
  - AES in counter (CTR) mode

Outline

- Random Numbers
  - DRBG
  - CTR_DRBG
  - Dual_EC_DRBG
- Stream Ciphers
- One Time Pad
- Visual Cryptography
- Prominent Stream Ciphers
  - A5 Cipher
  - RC4 Cipher
  - ChaCha Cipher
- Conclusion

Outline

- Random Numbers
  - DRBG
  - CTR_DRBG
  - Dual_EC_DRBG
- Stream Ciphers
- One Time Pad
- Visual Cryptography
- Prominent Stream Ciphers
  - A5 Cipher
  - RC4 Cipher
  - ChaCha Cipher
- Conclusion

Randomness

- Defining randomness is difficult.
- What we want: any specific string of bits is exactly as random as any other string.
- *Generators* of random strings:
  - *True random number generator* (TRNG) is a physical process which outputs each valid string independently with equal probability.
  - *Pseudorandom number generator* (PRNG) is a deterministic algorithm which approximates a TRNG.
- Using a TRNG to provide a *seed* for a PRNG.

True Random Number Generator (TRNG)

- NIST Special Publication 800-90B (Jan. 2016):
  - Framework for design and validation of TRNG algorithms, called *entropy sources*.
  - Specification of statistical tests for validating the suitability of entropy sources.
- The entropy source includes:
  - A physical noise source
  - A digitization process
  - Post-processing stages
- The output of the entropy source is any requested number of bits.
- Periodic *health test* to ensure continuing reliable operation.
- Intel introduced TRNG into Ivy Bridge processors in 2012.

Pseudorandom Number Generator (PRNG)

- NIST Special Publication 800-90A (June 2015):
  - Recommendation of specific PRNG algorithms, named *deterministic random bit generator* (DRBG)
  - DRBG is based on hash functions, a specific MAC (known as HMAC) and block ciphers in counter mode.
- Each generator takes a seed as input.
- It outputs a bit string before updating its state.
- The seed should be updated after some number of calls.
- The seed can be obtained from a TRNG.

Functions

- *Instantiate:* setting the initial state of the DRBG using a seed.
- *Generate:* providing an output bit string for each request.
- *Reseed:* inputting a new random seed and updating the state.
- *Test:* checking correct operation of the other functions.
- *Uninstantiate:* deleting (zeroising) the state of the DRBG.

Security

Security w.r.t. the ability of an attacker to distinguish reliably between its output and a truly random string:

- *Backtracking resistance:* an attacker who obtains the current state of the DRBG should not be able to distinguish between the output of *earlier calls* to the function Generate and random strings.
- *Forward prediction resistance:* an attacker who obtains the current state of the DRBG should not be able to distinguish between the output of *later calls* to the function Generate and random strings.

CTR_DRBG

- Using a block cipher in counter (CTR) mode:
  - Recommendation: AES with 128-bit keys
- DRBG initialised with a seed whose length is equal to the key length PLUS the block length:
  - $128 + 128 = 256$ for AES with 128-bit master keys
- Seed defines a key K and a counter value ctr:
  - No separate nonce as in a normal CTR mode
- CTR mode encryption is run iteratively, with no plaintext added.
- The output blocks form the CTR_DRBG output.

Update Function

- Each request to DRBG generates up to 2^{19} bits.
- From the function Generate:
  - (K, ctr)'s state must be updated after each request by generating 2 blocks using the current key to obtain the new key and a counter.
- Updating provides backtracking resistance.
- Restriction: up to 2^{48} requests to the function Generate before requiring re-seeding.
- Each re-seed provides forward prediction and backtracking resistance.

Dual_EC_DRBG

- From an older standard (Dec. 2012).
- Based on elliptic curve discrete logarithm problem:
  - But no security proof exists
  - And many flaws:
    `https://blog.cryptographyengineering.com/2013/09/18/the-many-flaws-of-dualecdrbg/`
- Much slower than other DRBGs in the standard.
- Press (Reuters) reported a secret 10 million dollar deal between NSA and RSA Security company to use Dual_EC_DRBG as the default PRNG in its software suite (Dec. 2013).

Outline

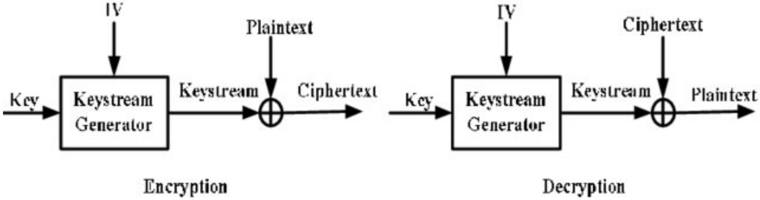
- Random Numbers
  - DRBG
  - CTR_DRBG
  - Dual_EC_DRBG
- Stream Ciphers
- One Time Pad
- Visual Cryptography
- Prominent Stream Ciphers
  - A5 Cipher
  - RC4 Cipher
  - ChaCha Cipher
- Conclusion

Stream Ciphers

- Characterised by the generation of a *keystream* using a short key and an initialisation value IV.
- Each element of the keystream is used successively to encrypt 1 or more ciphertext characters.
- Usually symmetric key ciphers:
  - The sender and receiver share the same key.
  - They can generate the same keystream given the same IV.

Synchronous Stream Ciphers

- The keystream is generated *independently* of the plaintext.
- Both sender and receiver need to generate the same keystream and synchronise on its usage.
- Vigenère cipher seen as a (periodic) synchronous stream cipher where each shift is defined by a key letter.
- CTR mode of operation for a block cipher is one method to generate a keystream.



Binary Synchronous Stream Ciphers

For each time interval t:

- Binary sequence $s(t)$, that is the keystream
- Binary plaintext $p(t)$
- Binary ciphertext $c(t)$

Encryption: $c(t) = p(t) \oplus s(t)$
Decryption: $p(t) = c(t) \oplus s(t)$

Outline

- Random Numbers
  - DRBG
  - CTR_DRBG
  - Dual_EC_DRBG
- Stream Ciphers
- One Time Pad
- Visual Cryptography
- Prominent Stream Ciphers
  - A5 Cipher
  - RC4 Cipher
  - ChaCha Cipher
- Conclusion

One Time Pad

- Often attributed to Vernam who made a one-time pad machine using teletype machinery in 1917 (earlier historical uses are known).
- Key is a random sequence of characters s.t. all of them are independently generated.
- Each character in the key is used ONE TIME ONLY.
- Alphabet of any length but usually:
  - A natural language alphabet
  - The binary alphabet $\{0, 1\}$
- Example: (non-periodic) binary synchronous stream cipher.
- Providing *perfect secrecy*.

Perfect Secrecy

Shannon's definition:

- Message set $\{M_1, \cdots, M_k\}$.
- Ciphertext set $\{C_1, \cdots, C_l\}$.
- $\Pr(M_i|C_j)$ is the probability that M_i is encrypted given that C_j is observed.
- In most cases, the messages M_i are NOT be equally likely.
- For all messages M_i and ciphertexts C_j:

$$\Pr(M_i|C_j) = \Pr(M_i)$$

One Time Pad Using Roman Alphabet

- Plaintext characters: $p_1, \cdots, p_r$
- Ciphertext characters: $c_1, \cdots, c_r$
- Keystream: random characters $k_1, \cdots, k_r$
- Encryption: $c_i = (p_i + k_i) \mod 26$
- Decryption: $p_i = (c_i - k_i) \mod 26$
- Ciphertext is the addition of plaintext and keystream characters, modulo 26.

One Time Pad Perfect Secrecy

- Let a ciphertext C_j be observed.
- Any message could have been sent, depending on the keystream.
- The probability that M_i is sent given that C_j is observed = the probability that M_i is chosen, weighted by the probability that the right keystream is chosen.
- Each key is chosen with equal probability.
- Conditional probability is thus $\Pr(M_i|C_j) = \Pr(M_i)$

Example

- Plaintext: HELLO
- Keystream: EZABD
- Ciphertext: LDLMR
- Given the ciphertext, the plaintext can be ANY 5-letter message.

Vernam Binary One Time Pad

- Plaintext: binary sequence $b_1, \cdots, b_r$
- Ciphertext: binary sequence $c_1, \cdots, c_r$
- Keystream: random binary sequence $k_1, \cdots, k_r$
- Encryption: $c_i \equiv p_i \oplus k_i$
- Decryption: $p_i \equiv c_i \oplus k_i$
- Keystream is SAME length as plaintext.
- Providing perfect secrecy since any ciphertext is equally possible given the plaintext.
- Encryption and decryption are identical processes.

Properties

- Shannon showed that any cipher with perfect secrecy MUST have as many keys as there are messages.
- One time pad is the ONLY unbreakable cipher.
- Practical usage is possible for pre-assigned communications between fixed parties.
- Problem: how to deal with key management of completely random keys?
  - Key generation, key transportation, key synchronization, key destruction are ALL problematic since the keys are SO large.

Outline

Random Numbers
- DRBG
- CTR_DRBG
- Dual_EC_DRBG

Stream Ciphers

One Time Pad

Visual Cryptography

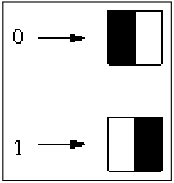
Prominent Stream Ciphers
- A5 Cipher
- RC4 Cipher
- ChaCha Cipher

Conclusion

Visual Cryptography

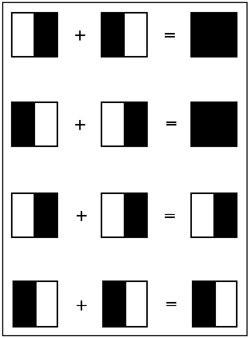
- Application of one time pad: visual cryptography splits an image into 2 shares.
- Decryption works by *overlaying* the 2 shared images.
- First proposed by Naor and Shamir in 1994.
- Simple case: monochrome images with black and white pixels.
- Many generalisations are possible.
- Each pixel is shared in a random way, similar to splitting a bit in the one time pad.
- Each share reveals NO information about the image:
  - Unconditional security as one time pad.

Encryption



- Generate a one time pad P (random bit string) with length equal to the number of pixels for the image I
- Generate a share $S_{I,1}$ by replacing each bit in P using the sub-pixel patterns shown on the left
- Generate the other share $S_{I,2}$ s.t.:
  - the same as $S_{I,1}$ for all the white pixels of I
  - the opposite of $S_{I,1}$ for all black pixels of I

Decryption



- To reveal the hidden image I, $S_{I,1}$ and $S_{I,2}$ are overlayed
- Each black pixel of I is black in the overlay
- Each white pixel of I is half white in the overlay

Outline

Random Numbers
- DRBG
- CTR_DRBG
- Dual_EC_DRBG

Stream Ciphers

One Time Pad

Visual Cryptography

Prominent Stream Ciphers
- A5 Cipher
- RC4 Cipher
- ChaCha Cipher

Conclusion

A5 Cipher

- Binary synchronous stream cipher applied in most GSM mobile telephones.
- 3 variants:
  - A5/1 is the original algorithm defined in 1987.
  - A5/2 is a *weakened* version of A5/1, originally intended for deployment outside Europe, but no longer allowed under GSM standards.
  - A5/3, also known as KASUMI, is an algorithm for deployment in 3G mobile systems.
- Design was originally kept confidential by its designers but became public in 1994.

A5/1 Design

- A5/1 algorithm uses 3 linear feedback shift registers (LFSRs) whose output is combined.
- The 3 LFSRs are *irregularly clocked*:
  - The overall output is non-linear.
  - 64-bit keystream s.t. 10 bits fixed at zero.
  - The effective key length is thus 54 bits.
- Many successful attacks:
  - In 2008, Gendrullis, Novotny and Rupp reported an attack which broke A5/1 in practice in 7 hours given.

RC4 Cipher

- World-based stream cipher designed by Ron Rivest in the 80s: “Ron’s code $\#$4”.
- Simple, efficient for software implementation.
- Originally proprietary owned by RSA Security, but leaked in 1994.
- Widely deployed in TLS before 2013.
- Practical attacks:
  - When used in TLS protocol and in wireless WPA-TKIP due to bias in its keystream output.
- Widely believed to be too weak to use in new systems.

ChaCha Algorithm

- Available in TLS ciphersuites (RFC 7905) as a possible replacement for RC4.
- Designed by D. J. Bernstein in 2008.
- Faster than AES:
  - As little as 4 cycles per byte on x86 processors.
- Combining XOR, addition modulo 2^{32} and rotation operations over 20 rounds to produce 512 bits of keystream:
  - Example: add-rotate-xor (ARX) cipher.
- Using 256-bit key.

Outline

Random Numbers
- DRBG
- CTR_DRBG
- Dual_EC_DRBG

Stream Ciphers

One Time Pad

Visual Cryptography

Prominent Stream Ciphers
- A5 Cipher
- RC4 Cipher
- ChaCha Cipher

Conclusion

Conclusion

- TRNG constructed from physical devices, used as seeds for PRNG.
- PRNG constructed from other primitives including block ciphers.
- TRNG used to make unbreakable encryption via one time pad.
- PRNG used as practical synchronous stream cipher.