## CPSC 418/MATH 318 Introduction to Cryptography

More Cryptanalysis of Block Ciphers, Stream Ciphers, Modes of Operation, One-Way Functions, Diffie-Hellman Key Exchange


## Outline

(1) Attacks on Block Ciphers

- Analytic Attacks
(2) Stream Ciphers
- Synchronous Stream Ciphers)
- Self-Synchronizing Stream Cipher)Modes of Operation for Block CiphersOne-Way Functions
(5)

Toward Cryptographic Key Agreement - Number Theory

- Primitive Roots and Discrete LogarithmsDiffie-Hellman Protocol


## Differential cryptanalysis

Biham and Shamir, Journal of Cryptology, 1991 — KPA
Compares input XORs to output XORs, and traces these differences through the cipher.

Both linear and differential cryptanalysis work quite well on DES with fewer than 16 rounds.

- The first edition of Doug Stinson's book "Cryptography - Theory and Practice" (1995) discusses successful differential cryptanalysis attacks on 3 -round and 6 -round DES.
- Large-scale, parallel, brute-force attack is still the most practical attack on 16 -round DES.

DES was designed to be resistant against differential cryptanalysis ("T" or "Tickle" attack). IBM and NSA knew about differential cryptanalysis.

Attacks on Block Ciphers Analytic Attacks

## Linear Cryptanalysis

## M. Matsui, EUROCRYPT 1993 - CCA

- Matsui actually used this method to become the first person to recover a DES key ( 50 days using 12 workstations).


## Definition 1

A cryptosystem is affine (linear) if encryptions are affine (linear) functions relating plaintexts to ciphertexts.

$$
\begin{array}{ll}
\text { Affine equation: } & C=A M+B \\
\text { Linear equation: } & C=A M \quad \text { (i.e. } B=0)
\end{array}
$$

where $A$ and $B$ are matrices of appropriate dimensions.
Idea: $A$ and $B$ reveal information about the key used to encrypt $M$ to $C$.

## Attacking Affine and Linear Cryptosystems

A cryptanalyst can try to mount a CPA on an affine or linear system by obtaining sufficiently many plaintext/ciphertext pairs $\left(M_{i}, C_{i}\right)$ to deduce $A$ and $B$ from the equations

$$
C_{i}=A M_{i}+B, \quad i=1,2,3, \ldots
$$

Examples of linear and affine cipher building blocks:

- transpositions - linear
- SubBytes operation in AES - affine on bytes

See Section 4.3.3 of Stinson-Paterson for an actual linear attack on a small substitution-permutation network.

Attacks on Block Ciphers Analytic Attacks

## Requirements for full DES

| Type of attack | Expected time | $\#$ of $(M, C)$ pairs |
| ---: | :---: | :---: |
| Exhaustive search | $2^{55}$ | none |
| Linear Cryptanalysis | $2^{43}$ | $2^{43}($ chosen $)$ |
| Differential Cryptanalysis | $2^{47}$ | $2^{47}($ known $)$ |

In DES, $2^{47}(M, C)$ pairs require 1 Petabyte ( $\approx 1,000$ Terrabytes) of storage.

Note: AES not affected by these attacks (by design).

Modern ciphers must be designed to credibly withstand linear and differential cryptanalysis!

## Idea of Linear Cryptanalysis

Linear cryptanalysis attempts to choose ( $M, C$ ) pairs such that with high probability, linear relations exist between portions of the plaintexts $M$ and ciphertexts $C$ (called "linearly approximations").

If a cryptosystem is "close to" being linear, then the modified system can be broken and the original system compromised after some searching.

- "close to linear" means modifying a few entries in the system (e.g. in the $S$-boxes) makes it linear on certain plaintext/ciphertext pairs.

Since $P$-boxes are linear, S-boxes must not be linear.

- S-boxes must also not be "close" to linear (i.e. closely approximated by a linear function).

DES was not designed to offer optimal resistance to linear cryptanalysis. Unclear if NSA or IBM did not know about linear cryptanalysis at the time or were just not worried about it.

## Attacks on Block Ciphers Analytic Attacks

## Algebraic Attacks

Courtois 2001 - KPA, generates multivariate equations from S-boxes, where the unknowns are the key bits.

- So far no threat to any modern block cipher.

Obstacle: solving multivariate equations seems to be hard in practice.
(In fact so hard that there are cryptosystems whose security is based on the intractability of this problem!)

## Attacks on Block Ciphers Analytic Attacks <br> Biclique Attacks

Enhanced meet-in-the-middle attack using bicliques that map internal states to ciphertexts via subkeys.

First improved key recovery through the biclique attack on AES (Bogdanov, Khovratovich, Rechberger 2011):

| AES key length | Average exhaustive search | Biclique (expected) |
| :---: | :---: | :---: |
| 128 | $2^{127}$ | $2^{126.1}$ |
| 192 | $2^{191}$ | $2^{189.7}$ |
| 256 | $2^{255}$ | $2^{254.4}$ |

- These and other attacks (e.g. square attack) are successful on 8 and lower round AES.
- Biclique attacks have also been successfully mounted on some lightweight ciphers


## Stream Ciphers

In contrast to block ciphers, stream ciphers don't treat incoming plaintext blocks independently.

- Encryption $C_{i}$ of plaintext block $M_{i}$ depends on internal state of device.
- After encryption, the device changes state according to some rule.

Result: two occurrences of the same plaintext will usually not result in the same ciphertext.

Stream ciphers incorporate a key stream into encryption and decryption that is generated from the key. In practice, this is a pseudo-random sequence of bits. Blocks of key bits are x-or'ed with plaintext blocks for encryption, and the same blocks are $x$-or'ed with ciphertext blocks for decryption

## Attacks on Block Ciphers Analytic Attacks <br> Lightweight Cryptography

Lightweight ciphers are systems targeted to operate in constrained environments, such as

- Sensors
- Healthcare devices
- Distributed control systems
- Internet of Things (loT) devices

See the NIST lightweight crypto competition at
https://csrc.nist.gov/projects/lightweight-cryptography

- Round 1: March 2019 - March 2021 (32 out of 57 submissions)
- Round 2: March 2021 - February 2023 (10 submissions)
- Ascon family selected for standardization on February 7, 2023
- Standardization process currently in progress

Up next: Stream ciphers.

## Synchronous Stream Cipher (SSC)

Idea:

- State depends only on the previous state, not on the input $M_{i}$.
- Output $C_{i}$ depends only on $M_{i}$ and $i$, not on $M_{i-1}, M_{i-2}, \ldots$
- Implemented by boolean logic that should produce a pseudo-random sequence $R_{i}$ synchronized by the key (e.g. a shift register like in Problem 1 of Assignment 1).


## Example 2

The one-time pad can be interpreted as an SSC. The key stream consists of the key bits.


## Stream Ciphers Synchronous Stream Ciphers)

Properties of Block-Cipher Based SSCs

## Advantages:

- Only the encryption function of the block cipher is used (important for AES where decryption is slightly less efficient than encryption)
- The fact that the $i$-th ciphertext block does not depend on previous ciphertext or plaintext blocks allows for random-access encryption/decryption and parallelism


## Problems:

(1) No error propagation
(2) Loss of one character between sender and receiver destroys synchronization (no "memory" of history)

Idea:

- Send an initial key value $K S_{0}=I V$ to the receiver in the clear.
- Compute $K S_{i}=E_{K}\left(K S_{i-1}\right)$ and $C_{i}=M_{i} \oplus K S_{i}$.



## Self-Synchronizing Stream Cipher (Self-SSC)

## AKA asynchronous stream cipher

Idea:

- Similar to SSC, except the counter is replaced by a register containing the previous $k$ ciphertexts.
- Self-synchronizing after $k$ steps.
- Can also be implemented with a block cipher as above.
- Limited error propagation (k steps).


## Diagram of a Self-SSC



Modes of Operation for Block Ciphers
More Modes of Operation

To eliminate the shortcomings of ECB mode, additional modes of operation have been devised:

- Cipher Block Chaining (CBC)
- Counter (CTR)
- Cipher Feedback (CFB)
- Output Feedback (OFB)

The last three modes turn a block cipher into a stream cipher.
DES Certified Modes: ECB, CBC, and CFB; standardized as part of DES standardization process.

- CTR mode arose from concerns with CBC; standardized for AES.


## Modes of Operation

Block ciphers can be used in a number of different modes of operation, depending on the application.

## Definition 3 (Electronic code book (ECB) mode)

Blocks are encrypted sequentially, one at a time: $C_{i}=E_{K}\left(M_{i}\right)$, $i=1,2, \ldots$.

A block cipher used in ECB mode is essentiallv a substitution cipher (with all its weaknesses).


Microsoft's Office 365 Message Encryption still used ECB in October 2022: https://www.theregister.com/2022/10/14/microsoft_office_365_message_encryption/ Changed to AES256-CBC in June 2023, with AES128-ECB legacy option

## Modes of Operation for Block Ciphers <br> Cipher Block Chaining (CBC) Mode

Send initial random block $C_{0}=I V$ (e.g. a simple plaintext encrypted in ECB mode, such as $C_{0}=E_{K}(00 \cdots 000)$

Encryption: $C_{i}=E_{K}(\underbrace{M_{i} \oplus C_{i-1}}) \quad i=1,2, \ldots$
"Pre-whitening"
Decryption: $M_{i}=D_{K}\left(C_{i}\right) \oplus C_{i-1} \quad i=1,2, \ldots$
Note that this is not a stream cipher (X-OR with plaintext happens inside encryption).

## Modes of Operation for Block Ciphers <br> Diagram of CBC



CTR, CFB and OFB all turn a block cipher into a stream cipher by generating a pseudorandom key stream $K S_{i}$ using the encryption function as described earlier:

$$
K S_{i}=E_{K}(\text { some string }), \quad C_{i}=M_{i} \oplus K S_{i} .
$$

Argument of $E_{K}$ is

- a counter value in CTR mode (synchronous)
- previous ciphertext bits in CFB mode (self-synchronizing)
- previous key stream bits in OFB mode (synchronous)
(1) Varying $I V$ encrypts the same message differently.
(2) Repeated plaintexts will be encrypted differently in different repetitions.
(3) Plaintext errors propagate through the rest of encryption (good for message authentication, as last ciphertext block depends on all plaintext blocks)
(1) Limited error propagation in decryption: error from incorrect ciphertext modification in propagates only to the next block.

Widely used, but vulnerabilities have been discovered (eg. Vaudenay 2002 padding attack, SSL insertion attack).

SSC with key stream $K S_{i}=E_{K}\left(C T R_{i}\right)$ where $C T R_{i}$ is a counter of the same size as the plaintext block size.

- Subsequent values of the counter are computed via an iterating function - the FIPS recommendation is simply $C T R_{i+1}=C T R_{i}+1$ $\left(\bmod 2^{n}\right)$ assuming an $n$-bit counter.

Counter must be unique for each plaintext block that is ever encrypted under a given key, across all messages.

- keep count of \# of plaintext blocks encrypted under a given counter sequence
- use a new block cipher key before exceeding $2^{n}$ blocks ( $n$-bit blocks)


## Feedback Modes

The feedback modes also turn a block cipher into a stream cipher:
CFB (cipher feedback) mode:

- self-SSC.
- Simplest form, one register: $K S_{i}=E_{K}\left(C_{i-1}\right)$ (with $C_{0}=I V$ ).
- In general, $r$ cipher bits are fed back (for DES, $r=8$ and IV is at least 48 random bits, right-justified, padded with 0 's).

OFB (output feedback) mode:

- SSC as described earlier
- Simplest form, one register: $K S_{i}=E_{K}\left(K S_{i-1}\right)$ (with $K S_{0}=I V$ )
- In general, $r$ keystream bits are fed back

For both feedback modes, each cryptographic session requires a different
IV, but as always, these may be sent in the clear.
Renate Scheidler (University of Calgary) CPSC 418/MATH $318 \quad$ Week 5 24/43

One-Way Functions
One-Way Functions

## Definition 4 (One-way function)

A function $f$ that satisfies the following two properties:
(1) Ease of Computation: $f(x)$ is efficient to evaluate for a given $x$.
(2) One-Way Property: Given $y=f(x)$, it is computationally intractable to find $x$.

It is not known whether true one way functions exist, but several that are believed to be one-way are used in cryptography. Instead of the one-way property, "practical" one-way functions satisfy the following weaker property:

## Definition 5 (Pre-image resistance)

(2) Pre-image Resistance: Given $y=f(x)$, it is computationally infeasible to find $x$.

## Further Information

More information can be found at NIST's cryptographic standards and guidelines website:

- For block ciphers, see https://csrc.nist.gov/projects/block-cipher-techniques
- For more modes of operation, see https://csrc.nist.gov/projects/block-cipher-techniques/BCM

Relevant publications: NIST SP 800-38A - 800-38G.

Next: working toward cryptographic key agreement via one-way functions.

## Example - Cryptosystem

## Example 6

A secure cryptosystem provides a one-way function as follows. Define

$$
f: \mathcal{K} \rightarrow \mathcal{C} \text { via } f(x)=E_{x}(M),
$$

where $M$ is a fixed known plaintext and $x$ is a key.
Given $M$ and $C$ (KPA), it should computationally infeasible to find the key $x$.

## Example - Polynomials

## Example 7

If $p$ is a large prime $\left(\approx 2^{1024}\right)$ and $g$ a suitably chosen integer (more later), then the function

$$
f(x)=g^{x} \quad(\bmod p)
$$

seems to be a one-way function, provided $p-1$ has at least one large prime factor.

- $x$ is the discrete logarithm (modulo $p$ ) of $f(x)$ with respect to $g$.
- Computing $x$ given $f(x)$ and $g$ is known as the discrete logarithm problem (DLP).

This function forms the basis of many cryptographic protocols.

## One-Way Functions <br> Application: Access Control

Secure login via one-way functions: Computer stores a table

$$
\left(\text { user-id }_{i}, f\left(P_{i}\right)\right),
$$

containing user id's and images of passwords under a one-way function $f$.

- Safer than storing passwords in the clear.

When a user logs in, she submits her user-id and her password $P$.
The computer generates $f(P)$ and checks if (user-id, $f(P)$ ) is an entry in the password table.

- If yes, access is granted; if no, access is denied.

Anyone gaining access to the table is unable to obtain $P$ from $f(P)$.

## Key Agreement Problem

Toward Cryptographic Key Agreement - Number Theory Primitive Roots and Discrete Logarithms

## Primitive Roots

For any prime $p$

- $\mathbb{Z}_{p}=\{0,1,2, \ldots, p-1\}$ is the set of integers modulo $p$;
- $\mathbb{Z}_{p}^{*}:=\mathbb{Z}_{p} \backslash\{0\}=\{1,2, \ldots, p-1\}$.


## Theorem 1 (Fermat)

If $a$ is an integer and $p$ is a prime with $p \nmid a$, then $a^{p-1} \equiv 1(\bmod p)$.
What about smaller powers of a, i.e. $a^{0}, a^{1}, a^{2}, \ldots a^{p-2}(\bmod p)$ ?

## Definition 9 (Primitive Root)

For a prime $p$, a primitive root of $p$ is an integer $g \in \mathbb{Z}_{p}^{*}$ such that the smallest positive exponent $k$ with $g^{k} \equiv 1(\bmod p)$ is $p-1$.

Mathematically, $g$ is a generator of the multiplicative group $\mathbb{Z}_{p}^{*}$.
Renate Scheidler (University of Calgary) CPSC 418/MATH 318 Week 5

Toward Cryptographic Key Agreement - Number Theory Primitive Roots and Discrete Logarithms

## Example

Primitive roots yield the longest possible cycle of powers modulo $p$.

## Example 10

Is $a=3$ a primitive root of $p=7$ ?
By tabulating the powers of $a \bmod p$ we get
$3^{0} \equiv 1, \quad 3^{1} \equiv 3, \quad 3^{2} \equiv 2, \quad 3^{3} \equiv 6, \quad 3^{4} \equiv 4, \quad 3^{5} \equiv 5, \quad 3^{6} \equiv 1 \quad(\bmod 7)$.
(Sequence repeats at exponent 6 by Fermat's theorem.)

- Since 6 is the smallest power of 3 yielding 1,3 is a primitive root of 7 .
- 5 is also a primitive root of 7 (check that!)

There are no others (e.g. $2^{3}=1$, so 2 is not a primitive root of 7 ).

## Example

## Properties of Primitive Roots

## Example 11

Is $g=2$ a primitive root of $p=19$ ?
$p-1=18=2 \times 3^{2}$

$$
\begin{aligned}
& 2^{(19-1) / 2}=2^{9} \equiv 18 \not \equiv 1 \quad(\bmod 19) \\
& 2^{(19-1) / 3}=2^{6} \equiv 7 \not \equiv 1 \quad(\bmod 19) .
\end{aligned}
$$

Thus, 2 is a primitive root of 19 .
Only two modular exponentiations rather than the 16 required to compute $g^{2}, g^{3}, \ldots, g^{17}(\bmod 19)$.

## Discrete Logarithms

Let $p$ be a prime and $g$ a primitive root of $p$. Then for every $y \in \mathbb{Z}_{p}^{*}$, there exists a unique integer $x$ with $0 \leq x \leq p-2$ such that

$$
y \equiv g^{x} \quad(\bmod p)
$$

## Definition 13 (Discrete Logarithm)

The integer $x$ is the discrete logarithm (or index) of $y$ (with respect to $g$ modulo $p$ ).

Recall that the function $f(x)=g^{x}(\bmod p)$ is (believed to be) a one-way function. This means that the discrete logarithm problem (DLP) extracting discrete logs - is computationally hard since it is equivalent to computing a pre-image of a one-way function.

Every element of $\mathbb{Z}_{p}^{*}$ is a unique power of a primitive root of $p$ with exponent between 0 and $p-2$

$$
\mathbb{Z}_{p}^{*}=\left\{g^{0}, g^{1}, \ldots, g^{p-2} \quad(\bmod p)\right\} .
$$

## Example 12

Recall the previous example:
$3^{0} \equiv 1, \quad 3^{1} \equiv 3, \quad 3^{2} \equiv 2, \quad 3^{3} \equiv 6, \quad 3^{4} \equiv 4, \quad 3^{5} \equiv 5, \quad 3^{6} \equiv 1 \quad(\bmod 7)$

$$
\mathbb{Z}_{7}^{*}=\{1,2,3,4,5,6\}=\left\{3^{0}, 3^{2}, 3^{1}, 3^{4}, 3^{5}, 3^{3}\right\},
$$

where all the powers of 3 are taken modulo 7 .

## Diffie-Hellman Key Exchange: Idea

Alice and Bob wish to establish a common key for encryption over a public channel in such a way that an eavesdropper cannot determine the key.


THE SECRET KEY IS: Two locks locked together.
Eavesdropper gets two locked locks \& cannot open them.

Diffie and Hellman (1976) — still used today.
Alice and Bob agree on

- a large prime $p$,
- a primitive root $g$ of $p(1<g<p)$.

These quantities can be public.

Alice and Bob publicly agree on $p=23$ and $g=5$.

| Alice | Public channel | Bob |
| :---: | :---: | :---: |
| Selects $a=17$ |  | Selects $b=12$ |
| $y_{a} \equiv 5^{17} \equiv 15(\bmod 23)$ | $\stackrel{15}{\longrightarrow}$ | 15 |
| 18 | $\leftarrow$ | $y_{b} \equiv 5^{12} \equiv 18(\bmod 23)$ |
| $K \equiv 18^{17} \equiv 8(\bmod 23)$ |  | $K \equiv 15^{12} \equiv 8(\bmod 23)$ |

The shared number is $K=8$.

Diffie-Hellman Description

| Alice | Public channel | Bob |
| :---: | :---: | :---: |
| Selects random $a$ <br> $(1<a<p-1)$ |  | Selects random $b$ <br> $(1<b<p-1)$ |
| $y_{a} \equiv g^{a}(\bmod p)$ | $\xrightarrow{y_{a}}$ | $y_{a}$ |
| $y_{b}$ | $\stackrel{y_{b}}{\longleftrightarrow}$ | $y_{b} \equiv g^{b}(\bmod p)$ |
| $K \equiv y_{b}{ }^{a}(\bmod p)$ |  | $K \equiv y_{a}{ }^{b}(\bmod p)$ |

## Note

$A$ and $B$ get the same number $K$ because

$$
y_{b}{ }^{a} \equiv\left(g^{b}\right)^{a} \equiv g^{b a} \equiv\left(g^{a}\right)^{b} \equiv y_{a}^{b} \quad(\bmod p)
$$

In practice, one could use the low order 128 bits of $H(K)$ for an AES key, where $H$ is a cryptographically secure hash function (more later).

How secure is this?

- How difficult is for an eavesdropper it to find $K$ ?
- In general, how should $p$ and $g$ be chosen to maximize security?

How efficient is this?

- How easy is it to find suitable values for $p$ and $g$ ?
- How long does it take to compute $y_{a} \equiv g^{a}(\bmod p)$ from $g$ and $a$ (also $y_{b}$ and $K$ )?

To answer these questions, we need more number theory!

