CSci 4271W
Development of Secure Software Systems
Day 17: Cryptography part 1: primitives

Stephen McCamant (he/him)

University of Minnesota, Computer Science & Engineering

Cryptography

The goal of cryptography is to provide a "secure channel" between two (or more) parties:

1. Revealing no information about the messages
2. Delivering only messages from Ape and Bear
3. Delivering messages in order or not at all.

Cryptography

The goal of cryptography is to provide a "secure channel" between two (or more) parties:

Cryptography

The goal of cryptography is to provide a "secure channel" between two (or more) parties:

Cryptography

The goal of cryptography is to provide a "secure channel" between two (or more) parties:

Even though Eve can inspect, modify, or drop any message and even if she knows there are only two possible conversations.

Keys

If Ape and Bear and Eve all know the same things, what keeps Eve from reading messages like Bear does?

Keys

If Ape and Bear and Eve all know the same things, what keeps Eve from reading messages like Bear does?

Bear knows a secret "key" that changes the decryption. Knowing it lets Ape and Bear keep secrets from Eve.

Caesar's cipher

To encrypt a secret message,

1. Write down each plaintext letter

```
ATTACK AT DAWN
```

2. Count ahead three letters from each plaintext letter to get a ciphertext letter.

```
ATTACK AT DAWN
BUUBDL BU EBXO
CVVCEM CV FCYP
DWWDFN DW GDZQ
```

Caesar's cipher

To encrypt a secret message,

1. Write down each plaintext letter

```
ATTACK AT DAWN
```

2. Count ahead three K letters from each plaintext letter to get a ciphertext letter.

```
ATTACK AT DAWN
BUUBDL BU EBXO
CVVCEM CV FCYP
DWWDFN DW GDZQ
```

More keys = more security?

```
  ATTACKATDAWN
+ CCCCCCCCCCCC
= CVVCEMCVFCYP

  ATTACKATDAWN
+ CATCATCATCAT
= CTMCCDCTWCWG
```

Poly-alphabetic shift cipher:

Rotate char 1 by K_1,

rotate char 2 by $K_2, \ldots$

rotate char n by K_n,

rotate char $n+1$ by $K_1, \ldots$

How many keys? $26^n \approx 10^{1.4n} \approx 2^{4.7n}$

More keys = more security?

```
  ATTACKATDAWN
+ CCCCCCCCCCCC
= CVVCEMCVFCYP

  ATTACKATDAWN
+ CATCATCATCAT
= CTMCCDCTWCWG
```

Poly-alphabetic shift cipher:

Rotate char 1 by K_1,

rotate char 2 by $K_2, \ldots$

rotate char n by K_n,

rotate char $n+1$ by $K_1, \ldots$

How many keys? $26^n \approx 10^{1.4n} \approx 2^{4.7n}$

Is it hard to break?

More keys = more security?

```
  ATTACKATDAWN
+ CCCCCCCCCCCC
= CVVCEMCVFCYP

  ATTACKATDAWN
+ CATCATCATCAT
= CTMCCDCTWCWG
```

Poly-alphabetic shift cipher:

Rotate char 1 by K_1,

rotate char 2 by $K_2, \ldots$

rotate char n by K_n,

rotate char $n+1$ by $K_1, \ldots$

How many keys? $26^n \approx 10^{1.4n} \approx 2^{4.7n}$

Is it hard to break? Not if $n = |K| \ll |M|$

More keys = more security?

```
  ATTACKATDAWN
+ CCCCCCCCCCCC
= CVVCEMCVFCYP

  ATTACKATDAWN
+ CATCATCATCAT
= CTMCCDCTWCWG
```

Poly-alphabetic shift cipher:

Rotate char 1 by K_1,

rotate char 2 by $K_2, \ldots$

rotate char n by K_n,

rotate char $n+1$ by $K_1, \ldots$

How many keys? $26^n \approx 10^{1.4n} \approx 2^{4.7n}$

Is it hard to break? Not if $n = |K| \ll |M|$

Special case when $|M| = |K|$: the one-time pad

```
RJZJVLSSRFKNZJCTYZTSIAIWKTRSEQIYCTYISBR
QIYIUKRRQEJMYIBSXYSRHZHVJSQRDPHXBSXHRAQ
PHXHTJQQPDILXHARWXRQGYGUIRPQCOGWARWGQZP
OGWGSIPPOCHKWGZQVWQPFXFTHQOPBNFVZQVFPYO
NFVFRHOONBGJVFYPUVPOEWESGPNOAMEUYPUEOXN
MEUEQGNNMAFIUEXOTUONDVDRFOMNZLDTXOTDNWM
LDTDPFMMLZEHTDWNSTNMCUCQENLMYKCSWNSCMVL
KCSCOELLKYDGSCVMRSMLBTBPDMKLXJBRVMRBLUK
JBRBNDKKJXCFRBULQRLKASAOCLJKWIAQULQAKTJ
IAQAMCJJIWBEQATKPQKJZRZNBKIJVHZPTKPZJSI
```

```
RJZJVLSSRFKNZJCTYZTSIAIWKTRSEQIYCTYISBR
QIYIUKRRQEJMYIBSXYSRHZHVJSQRDPHXBSXHRAQ
PHXHTJQQPDILXHARWXRQGYGUIRPQCOGWARWGQZP
OGWGSIPPOCHKWGZQVWQPFXFTHQOPBNFVZQVFPYO
NFVFRHOONBGJVFYPUVPOEWESGPNOAMEUYPUEOXN
MEUEQGNNMAFIUEXOTUONDVDRFOMNZLDTXOTDNWM
LDTDPFMMLZEHTDWNSTNMCUCQENLMYKCSWNSCMVL
KCSCOELLKYDGSCVMRSMLBTBPDMKLXJBRVMRBLUK
JBRBNDKKJXCFRBULQRLKASAOCLJKWIAQULQAKTJ
IAQAMCJJIWBEQATKPQKJZRZNBKIJVHZPTKPZJSI
```

```
RJZJVLSSRFKNZJCTYZTSIAIWKTRSEQIYCTYISBR
QIYIUKRRQEJMYIBSXYSRHZHVJSQRDPHXBSXHRAQ
PHXHTJQQPDILXHARWXRQGYGUIRPQCOGWARWGQZP
OGWGSIPPOCHKWGZQVWQPFXFTHQOPBNFVZQVFPYO
NFVFRHOONBGJVFYPUVPOEWESGPNOAMEUYPUEOXN
MEUEQGNNMAFIUEXOTUONDVDRFOMNZLDTXOTDNWM
LDTDPFMMLZEHTDWNSTNMCUCQENLMYKCSWNSCMVL
KCSCOELLKYDGSCVMRSMLBTBPDMKLXJBRVMRBLUK
JBRBNDKKJXCFRBULQRLKASAOCLJKWIAQULQAKTJ
IAQAMCJJIWBEQATKPQKJZRZNBKIJVHZPTKPZJSI
```

```
RJZJVLSSRFKNZJCTYZTSIAIWKTRSEQIYCTYISBR
QIYIUKRRQEJMYIBSXYSRHZHVJSQRDPHXBSXHRAQ
PHXHTJQQPDILXHARWXRQGYGUIRPQCOGWARWGQZP
OGWGSIPPOCHKWGZQVWQPFXFTHQOPBNFVZQVFPYO
N V R O N G V Y U P E E G N A E Y U O N
 E E G N A I E O U N V R O N L T O D W
LDTDPFMMLZEHTDWNSTNMCUCQENLMYKCSWNSCMVL
KCSCOELLKYDGSCVMRSMLBTBPDMKLXJBRVMRBLUK
JBRBNDKKJXCFRBULQRLKASAOCLJKWIAQULQAKTJ
IAQAMCJJIWBEQATKPQKJZRZNBKIJVHZPTKPZJSI
```

Big numbers

Modern cryptography is based on the idea of making the best (known) algorithms take too much time to be feasible.

2^{30}	$\sim$ minutes on your phone
2^{40}	$\sim$ hours on a laptop
2^{50}	\$100 on AWS
2^{70}	$<$ 1 hour of (global) cryptocurrency mining output
2^{80}	safe for a year or two?
2^{128}	warm fuzzy cryptographer happiness zone
2^{256}–2^{1024}	#atoms in the universe – #bit ops until heat death

Outline

Cryptography problem statement and history

Announcements intermission

Common cryptographic primitives

Upcoming assignments

- Homework 5 (cryptography) out today, due next Tuesday night
  - Questions 1 and 2 relate to today's lecture material
- Project 2 materials now available (sorry for delays)
  - Now also available: more details on writing rubric
  - Individual section drafts due next Thursday 4/10
  - Due in two weeks, Tuesday 4/15

Outline

Cryptography problem statement and history

Announcements intermission

Common cryptographic primitives

Symmetric cryptography

In symmetric crypto, Ape and Bear know the same key K_{ab}.

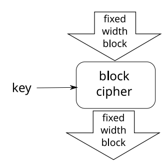
Ape computes ciphertext $= E_{K_{ab}}$(plaintext)
Bear recovers plaintext $= D_{K_{ab}}$(ciphertext)

Symmetric encryption

Comes in two forms, block ciphers and stream ciphers

Symmetric encryption

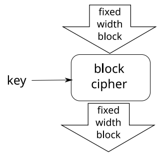
Comes in two forms, block ciphers and stream ciphers

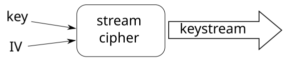


Ex: AES, 3DES, RC6
Needs to use a mode

Symmetric encryption

Comes in two forms, block ciphers and stream ciphers





Ex: AES, 3DES, RC6
Needs to use a mode

Ex: ~~RC4~~, Salsa, ChaCha20
Keystream XORed with plaintext

Symmetric authentication

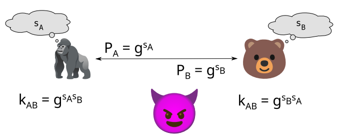
It's usually a mistake to use symmetric crypto that isn't also authenticated with a Message Authentication Code (MAC)

Any change to `ctxt` results in a new, unguessable tag.

Some block cipher modes do this automatically, e.g. AES-GCM, AES-CCM. See also ChaCha20-Poly1305

Asymmetric cryptography

In asymmetric (public-key) cryptography, each party has a secret key and a public key.



Asymmetric crypto is slower than symmetric, and requires longer keys. Normally used to share a symmetric key.

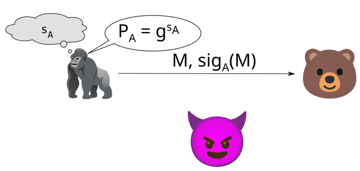
Asymmetric primitives

Asymmetric crypto algorithms are related to hard math (e.g., number theory) problems.

Algorithm	Problem	2^{80} work key	2^{128} key
RSA	factoring	1024 bits	3072 bits
Diffie-Hellman (Elgamal)	discrete log	1024 bits	3072 bits
ECDH	elliptic curve	160 bits	256 bits
Kyber (ML-KEM)	MLWE	~500 bytes	800 bytes

Signatures

Ape has a private signing key and a public verification key.



Only someone who knows s_A can make a valid $\text{sig}_A(M)$.
Signatures provide authenticity and non-repudiation.

Signature algorithms

Signature algorithms are related to hard math (e.g., number theory) problems.

Algorithm	Problem	2^{128} work key	Signature length
RSA	factoring	3072 bits	1024 bits
DSA	discrete log	3072 bits	320 bits
ECDSA	elliptic curve	256 bits	512 bits
EdDSA	elliptic curve	256 bits	512 bits
ML-DSA	MLWE	1312 bytes	2420 bytes

Hashing

Hash functions turn long strings into k-bit strings.
Finding a preimage $H(x) = y$ given y should take about:
Finding a targeted collision $H(x_1) = H(x_2)$ given x_2 takes:
Finding a free collision $H(x_1) = H(x_2)$ should take:
Examples of useful hash functions: SHA256+, SHA3, BLAKE
Winding down/insecure: MD5, SHA1, MD4, RIPEMD160

Hashing

Hash functions turn long strings into k-bit strings.
Finding a preimage $H(x) = y$ given y should take about: 2^k
Finding a targeted collision $H(x_1) = H(x_2)$ given x_2 takes:
Finding a free collision $H(x_1) = H(x_2)$ should take:
Examples of useful hash functions: SHA256+, SHA3, BLAKE
Winding down/insecure: MD5, SHA1, MD4, RIPEMD160

Hashing

Hash functions turn long strings into k-bit strings.

Finding a preimage $H(x) = y$ given y should take about: 2^k

Finding a targeted collision $H(x_1) = H(x_2)$ given x_2 takes: 2^k

Finding a free collision $H(x_1) = H(x_2)$ should take:

Examples of useful hash functions: SHA256+, SHA3, BLAKE

Winding down/insecure: MD5, SHA1, MD4, RIPEMD160

Hashing

Hash functions turn long strings into k-bit strings.

Finding a preimage $H(x) = y$ given y should take about: 2^k

Finding a targeted collision $H(x_1) = H(x_2)$ given x_2 takes: 2^k

Finding a free collision $H(x_1) = H(x_2)$ should take: $2^{k/2}$

Examples of useful hash functions: SHA256+, SHA3, BLAKE

Winding down/insecure: MD5, SHA1, MD4, RIPEMD160

Psuedorandom generators

Cryptography needs unguessable inputs.

Things not to use: serial number, Unix timestamp, process ID, hostname/address, C `rand`, Python `random`

More or less safe: `_rdrand`, `/dev/urandom`, `os.random`, `java.security.SecureRandom`