OSY.SSI[2015][4]

Notions of cryptography

This is not a Crypto course.

This is not a Crypto course.

Okay maybe just a little

Table of Contents

Cryptography

A definition

Cryptography is the (hopefully) clever use of *hard-to-solve problems*.

Cryptography

A definition

Cryptography is the (hopefully) clever use of *hard-to-solve problems*.

That being said,

Cryptography

A definition

Cryptography is the (hopefully) clever use of *hard-to-solve problems*.

That being said,

- Hard, in general, doesn't mean impossible

Cryptography

A definition

Cryptography is the (hopefully) clever use of *hard-to-solve problems*.

That being said,

- Hard, in general, doesn't mean impossible
- "Clever" is subjective

Cryptography

A definition

Cryptography is the (hopefully) clever use of *hard-to-solve problems*.

That being said,

- Hard, in general, doesn't mean impossible
- "Clever" is subjective
- It is rarely the weakest link

Cryptography

A definition

Cryptography is the (hopefully) clever use of *hard-to-solve problems*.

That being said,

- Hard, in general, doesn't mean impossible
- "Clever" is subjective
- It is rarely the weakest link

Nevertheless, it's interesting so let's talk about it.


SHUT
OPEN

Cryptography

Hard problems

Most “hard problems” people are interested with are really *math problems*.

Cryptography

Hard problems

Most “hard problems” people are interested with are really *math problems*.

Examples:

- Find the discrete logarithm of y base g modulo p ;
- Given z, g^x and g^y modulo p, tell whether $z = g^{xy}$;
- Find the shortest vector in an Euclidean lattice ;
- Find a number dividing x ;

How “hard” are they?

Cryptography

Hard problems

Most "hard problems" people are interested with are really *math problems*.

Examples:

- Find the discrete logarithm of y base g modulo p ;
- Given z, g^x and g^y modulo p, tell whether $z = g^{xy}$;
- Find the shortest vector in an Euclidean lattice ;
- Find a number dividing x ;

How "hard" are they?

In general, we mean that *there is no polynomial-time probabilitistic algorithm succeeding with non-negligible probability*.

Cryptography

Hard problems

Most "hard problems" people are interested with are really *math problems*.

Examples:

- Find the discrete logarithm of y base g modulo p ;
- Given z, g^x and g^y modulo p, tell whether $z = g^{xy}$;
- Find the shortest vector in an Euclidean lattice ;
- Find a number dividing x ;

How "hard" are they?

In general, we mean that *there is no polynomial-time probabilitistic algorithm succeeding with non-negligible probability*.

Note: if $\mathbf{P} = \mathbf{NP}$, then hard problems don't exist.

Cryptography

The framework

When using cryptography for security, we specify:

Cryptography

The framework

When using cryptography for security, we specify:

- The adversary model considered

Cryptography

The framework

When using cryptography for security, we specify:

- The adversary model considered
- A certain *game* he shouldn't win

Cryptography

The framework

When using cryptography for security, we specify:

- The adversary model considered
- A certain *game* he shouldn't win

Example:

Cryptography

The framework

When using cryptography for security, we specify:

- The adversary model considered
- A certain *game* he shouldn't win

Example:

$$\underbrace{\texttt{DotA2}}_{\text{Game}} - \underbrace{\texttt{Polard}}_{\text{Adversary}}$$

Cryptography

Shannon's game: IND-ONETIME

Cryptography

Shannon's game: IND-ONETIME

- *Setup*: Alice choses $b = 0$ or 1.

Cryptography

Shannon's game: IND-ONETIME

- *Setup*: Alice choses $b = 0$ or 1.
- *Challenge*: Bob sends two messages, m_0 and m_1.

Cryptography

Shannon's game: IND-ONETIME

- *Setup*: Alice choses $b = 0$ or 1.
- *Challenge*: Bob sends two messages, m_0 and m_1.
- *Response*: Alice encodes m_b and sends the result.

Cryptography

Shannon's game: IND-ONETIME

- *Setup*: Alice choses $b = 0$ or 1.
- *Challenge*: Bob sends two messages, m_0 and m_1.
- *Response*: Alice encodes m_b and sends the result.
- *Closure*: Bob sends a value b'.

Cryptography

Shannon's game: IND-ONETIME

- *Setup*: Alice choses $b = 0$ or 1.
- *Challenge*: Bob sends two messages, m_0 and m_1.
- *Response*: Alice encodes m_b and sends the result.
- *Closure*: Bob sends a value b'.

Bob wins the game if $b' = b$. Otherwise he loses.

Cryptography

Shannon's game: IND-ONETIME

- *Setup*: Alice choses $b = 0$ or 1.
- *Challenge*: Bob sends two messages, m_0 and m_1.
- *Response*: Alice encodes m_b and sends the result.
- *Closure*: Bob sends a value b'.

Bob wins the game if $b' = b$. Otherwise he loses.

The *advantage* is the probability that Bob wins the game:

$$\mathcal{A}^{\mathrm{IND}} = \left| \Pr\left[\text{Bob wins}\right] - \frac{1}{2} \right|$$

We want this to be as small as possible.

Cryptography

Shannon's game: IND-ONETIME

- *Setup*: Alice choses $b = 0$ or 1.
- *Challenge*: Bob sends two messages, m_0 and m_1.
- *Response*: Alice encodes m_b and sends the result.
- *Closure*: Bob sends a value b'.

Bob wins the game if $b' = b$. Otherwise he loses.

The *advantage* is the probability that Bob wins the game:

$$\mathcal{A}^{\mathrm{IND}} = \left| \Pr\left[\text{Bob wins}\right] - \frac{1}{2} \right|$$

We want this to be as small as possible.

Cryptography

Shannon's game: IND-ONETIME

$$
\begin{aligned}
m_0 &= 42 \\
m_1 &= 69 \\
X &= 1328111406936738814233136598523649120591057076609366993787434974992249717152208863269860655190296374487681635030289115548853940474812667441442229218578540925220642653758876912554424018330886760662928726528056192926786208742650591040002226597309870518868043425184899501098511877398934999057859295516670925844 29
\end{aligned}
$$

Which one is it?

Cryptography

Kerchoff's principle – Security by design

(Auguste Kerckhoffs, *La Cryptographie Militaire*, 1883.)

Cryptography

Kerchoff's principle – Security by design

(Auguste Kerckhoffs, *La Cryptographie Militaire*, 1883.)

Assume even the weakest adversary knows how your cryptographic primitives work.

Cryptography

Kerchoff's principle – Security by design

(Auguste Kerckhoffs, *La Cryptographie Militaire*, 1883.)

Assume even the weakest adversary knows how your cryptographic primitives work.

i.e. there is no "security through obscurity".

Cryptography

IND : adversary models

Cryptography

IND : adversary models

- ONETIME: the adversary only has one shot ;

Cryptography

IND : adversary models

- ONETIME: the adversary only has one shot ;
- CPA: the adversary may have several known plaintexts encrypted ;

Cryptography

IND : adversary models

- ONETIME: the adversary only has one shot ;
- CPA: the adversary may have several known plaintexts encrypted ;
- CCA: the adversay may have several known ciphertexts decrypted ;

Cryptography

IND : adversary models

- ONETIME: the adversary only has one shot ;
- CPA: the adversary may have several known plaintexts encrypted ;
- CCA: the adversay may have several known ciphertexts decrypted ;
- CCA2: the adversary may decrypt adaptively messages during the game.

Cryptography

IND : adversary models

- ONETIME: the adversary only has one shot ;
- CPA: the adversary may have several known plaintexts encrypted ;
- CCA: the adversay may have several known ciphertexts decrypted ;
- CCA2: the adversary may decrypt adaptively messages during the game.

Question: What is the security level of RSA ?

Table of Contents

What do we do with cryptography?

From a ITsec perspective

What do we do with cryptography?

From a ITsec perspective

One main and most ubiquitous use of crypto is to provide **integrity** guarantees.

What do we do with cryptography?

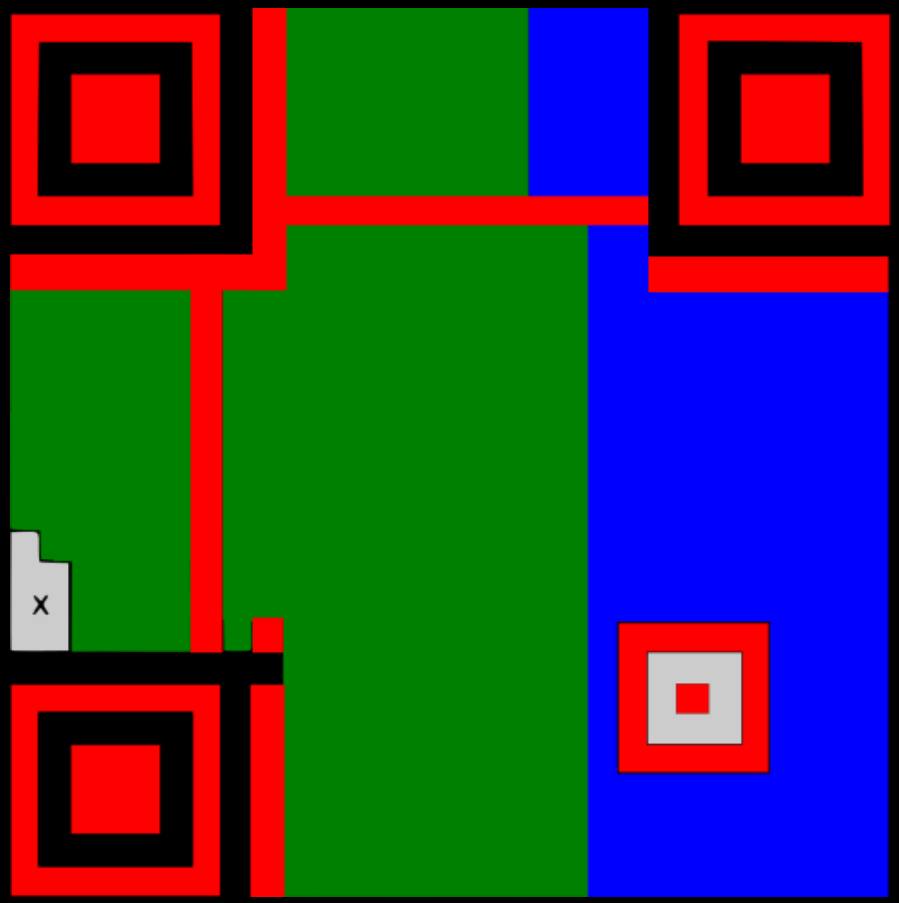
From a ITsec perspective

One main and most ubiquitous use of crypto is to provide **integrity** guarantees.

The whole branch of “error correcting codes” deals with $\perp$.

What do we do with cryptography?

Error correcting codes



Anatomy of a QR Code v3 level H.

What do we do with cryptography?

Message authentication codes

Alice sends a message to Bob. How does he know the message is correct ?

What do we do with cryptography?

Message authentication codes

Alice sends a message to Bob. How does he know the message is correct ?

- Error-correcting codes work only up to a point

What do we do with cryptography?

Message authentication codes

Alice sends a message to Bob. How does he know the message is correct ?

- Error-correcting codes work only up to a point
- An adversary could intercept the message, and generate another one, with the correct ECCs

What do we do with cryptography?

Message authentication codes

Alice sends a message to Bob. How does he know the message is correct ?

- Error-correcting codes work only up to a point
- An adversary could intercept the message, and generate another one, with the correct ECCs

Cryptography provides instead several MAC algorithms.

What do we do with cryptography?

Message authentication codes

Alice sends a message to Bob. How does he know the message is correct ?

- Error-correcting codes work only up to a point
- An adversary could intercept the message, and generate another one, with the correct ECCs

Cryptography provides instead several MAC algorithms.

MAC Algorithms uses a *secret key* to hash a message.

What do we do with cryptography?

Message authentication codes

Alice sends a message to Bob. How does he know the message is correct ?

- Error-correcting codes work only up to a point
- An adversary could intercept the message, and generate another one, with the correct ECCs

Cryptography provides instead several MAC algorithms.

MAC Algorithms uses a *secret key* to hash a message.

Question: assumptions? weaknesses?

What do we do with cryptography?

Message authentication codes

Alice sends a message to Bob. How does he know the message is correct ?

- Error-correcting codes work only up to a point
- An adversary could intercept the message, and generate another one, with the correct ECCs

Cryptography provides instead several MAC algorithms.

MAC Algorithms uses a *secret key* to hash a message.

Question: assumptions? weaknesses?
Question: encrypt and MAC? encrypt then MAC? MAC then encrypt?

What do we do with cryptography?

Message authentication codes

Alice sends a message to Bob. How does he know the message is correct ?

- Error-correcting codes work only up to a point
- An adversary could intercept the message, and generate another one, with the correct ECCs

Cryptography provides instead several MAC algorithms.

MAC Algorithms uses a *secret key* to hash a message.

Question: assumptions? weaknesses?
Question: encrypt and MAC? encrypt then MAC? MAC then encrypt?
Encrypt-then-MAC.

About those keys

Problem:

About those keys

Problem: we want to send an encrypted message to Bob.

About those keys

Problem: we want to send an encrypted message to Bob. Bob needs the key.

About those keys

Problem: we want to send an encrypted message to Bob. Bob needs the key.

Ideas?

A solution: (Diffie, Hellman, Merkle and GCHQ, 1970s)

About those keys

Problem: we want to send an encrypted message to Bob. Bob needs the key.

Ideas?

A solution: (Diffie, Hellman, Merkle and GCHQ, 1970s) "Public-key cryptography"!

About those keys

Problem: we want to send an encrypted message to Bob. Bob needs the key.

Ideas?

A solution: (Diffie, Hellman, Merkle and GCHQ, 1970s) "Public-key cryptography"!

- Alice chooses a secret x and sends $X = g^x$

About those keys

Problem: we want to send an encrypted message to Bob. Bob needs the key.

Ideas?

A solution: (Diffie, Hellman, Merkle and GCHQ, 1970s) "Public-key cryptography"!

- Alice chooses a secret x and sends $X = g^x$
- Bob chooses a secret y and sends $Y = g^y$

About those keys

Problem: we want to send an encrypted message to Bob. Bob needs the key.

Ideas?

A solution: (Diffie, Hellman, Merkle and GCHQ, 1970s) "Public-key cryptography"!

- Alice chooses a secret x and sends $X = g^x$
- Bob chooses a secret y and sends $Y = g^y$
- Alice computes the key $K = Y^x = g^{xy}$

About those keys

Problem: we want to send an encrypted message to Bob. Bob needs the key.

Ideas?

A solution: (Diffie, Hellman, Merkle and GCHQ, 1970s) "Public-key cryptography"!

- Alice chooses a secret x and sends $X = g^x$
- Bob chooses a secret y and sends $Y = g^y$
- Alice computes the key $K = Y^x = g^{xy}$
- Bob computes the key $K = X^y = g^{xy}$

About those keys

Problem: we want to send an encrypted message to Bob. Bob needs the key.

Ideas?

A solution: (Diffie, Hellman, Merkle and GCHQ, 1970s) "Public-key cryptography"!

- Alice chooses a secret x and sends $X = g^x$
- Bob chooses a secret y and sends $Y = g^y$
- Alice computes the key $K = Y^x = g^{xy}$
- Bob computes the key $K = X^y = g^{xy}$

"Diffie-Hellman key-exchange" (CDH, DDH).

About those keys

ECDHE_RSA stands for Elliptic Curve Diffie-Hellman key-exchange, signed with RSA.

And about this AES_128_GCM thing?

And about this AES_128_GCM thing?

- AES is the *cipher* (here, Rijndael *aka* AES)

And about this AES_128_GCM thing?

- AES is the *cipher* (here, Rijndael *aka* AES)
- 128 is the *blocksize* (AES is a *block cipher*)

And about this AES_128_GCM thing?

- AES is the *cipher* (here, Rijndael *aka* AES)
- 128 is the *blocksize* (AES is a *block cipher*)
- GCM is the *mode of operation* (how we deal with multiple blocks), here “Galois counter mode”

And about this AES_128_GCM thing?

- AES is the *cipher* (here, Rijndael *aka* AES)
- 128 is the *blocksize* (AES is a *block cipher*)
- GCM is the *mode of operation* (how we deal with multiple blocks), here "Galois counter mode"

More on that another time.

And about this AES_128_GCM thing?

- AES is the *cipher* (here, Rijndael *aka* AES)
- 128 is the *blocksize* (AES is a *block cipher*)
- GCM is the *mode of operation* (how we deal with multiple blocks), here "Galois counter mode"

More on that another time.

Just don't use ECB, CBC, CFB, OFB modes! (and avoid CTR if you can)

Two key issues with cryptography

For our purposes

1. How to *correctly use it* (what to encrypt, how, what mode, what blocksize?, where to sign, when to MAC? etc.)

Two key issues with cryptography

For our purposes

1. How to *correctly use it* (what to encrypt, how, what mode, what blocksize?, where to sign, when to MAC? etc.)
2. How to *correctly implement it*

Two key issues with cryptography

For our purposes

1. How to *correctly use it* (what to encrypt, how, what mode, what blocksize?, where to sign, when to MAC? etc.)
2. How to *correctly implement it*

Hint:

Two key issues with cryptography

For our purposes

1. How to *correctly use it* (what to encrypt, how, what mode, what blocksize?, where to sign, when to MAC? etc.)
2. How to *correctly implement it*

Hint: don't imitate Apple, Adobe, Microsoft, Sony or Snapchat...

Two key issues with cryptography

For our purposes

1. How to *correctly use it* (what to encrypt, how, what mode, what blocksize?, where to sign, when to MAC? etc.)
2. How to *correctly implement it*

Hint: don't imitate Apple, Adobe, Microsoft, Sony or Snapchat...

Crypto and security

- Put hard-to-solve problems where it matters:
  - Integrity, Confidentiality
- Also, superb mathematics and techniques $</$ad$>$.

Crypto and security

- Put hard-to-solve problems where it matters:
  - Integrity, Confidentiality
- Also, superb mathematics and techniques </ad>.
- Bad crypto leads to bad security (we'll see a demo)

Crypto and security

- Put hard-to-solve problems where it matters:
  - Integrity, Confidentiality
- Also, superb mathematics and techniques $</$ad$>$.
- Bad crypto leads to bad security (we'll see a demo)
- Weak crypto leads to weak security (stolen backdoors...)

Crypto and security

- Put hard-to-solve problems where it matters:
  - Integrity, Confidentiality
- Also, superb mathematics and techniques </ad>.
- Bad crypto leads to bad security (we'll see a demo)
- Weak crypto leads to weak security (stolen backdoors...)
- No crypto...

... next week: The Internet part I