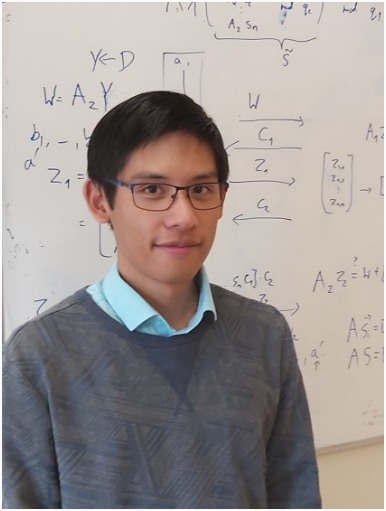
6CCS3CIS Cryptography, first lecture

Ngoc Khanh Nguyen and Eamonn W. Postlethwaite

September 11, 2025

Hello





Administrative

This is the third year 'level six' variant, with office hours by email (Eamonn 16–17 Mon, Khanh 13–14 Fri).

There are weekly lectures 16–18 Thurs and tutorials on Monday or Tuesday.

The grade is 100% written examination in Jan with a capped resit, a mock exam will be made available.

Please use the KEATS forum for all questions of course content and administration.

I (Eamonn) will give the first five weeks and Khanh the second five.

Literature

Some books that can be used to add further context and formalism to this course (or simply provide another attempt at explanation) are

- Katz–Lindell, Introduction to Modern Cryptography, 3rd edition (abbrev: KL-IMC)
- Mike Rosulek, The Joy of Cryptography,[1] (abbrev: R-JC)

[1] Current draft available here `https://joyofcryptography.com/`.

Literature

Some books that can be used to add further context and formalism to this course (or simply provide another attempt at explanation) are

- Katz–Lindell, Introduction to Modern Cryptography, 3rd edition (abbrev: KL-IMC)
- Mike Rosulek, The Joy of Cryptography,[1] (abbrev: R-JC)

Please do not feel obliged to buy them! (The library has at least one copy of KL-IMC.) They are not necessary, more like bonus material.

[1] Current draft available here `https://joyofcryptography.com/`.

Think. . .

If you were asked to describe your understanding of the study or role of cryptography

- in the academic setting,
- in the industrial or economic setting,
- in the cultural or societal setting,

how would you respond?

Some kinds of definitions

Oxford English: the art or practice of writing in code or cipher; the science of encryption.

Some kinds of definitions

Oxford English: the art or practice of writing in code or cipher; the science of encryption. (I think this is BAD.)

Some kinds of definitions

Oxford English: the art or practice of writing in code or cipher; the science of encryption. (I think this is BAD.)

Katz–Lindell: the study of mathematical techniques for securing digital information, systems and distributed computations against adversarial attacks.

Some kinds of definitions

Oxford English: the art or practice of writing in code or cipher; the science of encryption. (I think this is BAD.)

Katz–Lindell: the study of mathematical techniques for securing digital information, systems and distributed computations against adversarial attacks.

Rosulek: [the study of] controlling access to information.

Some kinds of definitions

Oxford English: the art or practice of writing in code or cipher; the science of encryption. (I think this is BAD.)

Katz–Lindell: the study of mathematical techniques for securing digital information, systems and distributed computations against adversarial attacks.

Rosulek: [the study of] controlling access to information.

Martin: to study the limits of computing.

Some kinds of definitions

Oxford English: the art or practice of writing in code or cipher; the science of encryption. (I think this is BAD.)

Katz–Lindell: the study of mathematical techniques for securing digital information, systems and distributed computations against adversarial attacks.

Rosulek: [the study of] controlling access to information.

Martin: to study the limits of computing.

Eamonn: the study of how to secure computations, and perhaps more importantly, what it means to secure computations.

Some kinds of definitions

Oxford English: the art or practice of writing in code or cipher; the science of encryption. (I think this is BAD.)

Katz–Lindell: the study of mathematical techniques for securing digital information, systems and distributed computations against adversarial attacks.

Rosulek: [the study of] controlling access to information.

Martin: to study the limits of computing.

Eamonn: the study of how to secure computations, and perhaps more importantly, what it means to secure computations.

Wilde: 'to define is to limit.'

Some kinds of definitions

Oxford English: the art or practice of writing in code or cipher; the science of encryption. (I think this is BAD.)

Katz–Lindell: the study of mathematical techniques for securing digital information, systems and distributed computations against adversarial attacks.

Rosulek: [the study of] controlling access to information.

Martin: to study the limits of computing.

Eamonn: the study of how to secure computations, and perhaps more importantly, what it means to secure computations.

Wilde: 'to define is to limit.'

Blanchette: cryptography is code created with the sole purpose of regulating behaviour.

Communications

Often when asked what their understanding of cryptography is, people converge towards the idea of secure communications.

Communications

Often when asked what their understanding of cryptography is, people converge towards the idea of secure communications. This certainly has the most historical precedence!

Communications

Often when asked what their understanding of cryptography is, people converge towards the idea of secure communications. This certainly has the most historical precedence!







A thought experiment

Two lords, Francis and Henry, in their pavilions want to communicate ‘securely’. In person, they decide between themselves

A thought experiment

Two lords, Francis and Henry, in their pavilions want to communicate ‘securely’. In person, they decide between themselves

- an encryption method that uses a *secret key* to transform *message* into *ciphertext*,
- a decryption method that uses the same *secret key* to transform *ciphertext* into its underlying *message*,
- a *secret key* (they whisper it to each other to make sure no one hears).

A thought experiment

Two lords, Francis and Henry, in their pavilions want to communicate 'securely'. In person, they decide between themselves

- an encryption method that uses a *secret key* to transform *message* into *ciphertext*,
- a decryption method that uses the same *secret key* to transform *ciphertext* into its underlying *message*,
- a *secret key* (they whisper it to each other to make sure no one hears).

When Francis wants to communicate a *message* with Henry he

- encrypts as above to create a *ciphertext* and writes it onto parchment,
- gives it to his most trusted rider to deliver to Henry,
- Henry then decrypts as above and reads the *message*.

Secret key encryption

We will return to this ad nauseam, but we just described a secret key encryption scheme (SKE). This is a triple of algorithms $\mathsf{SKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ such that

- KGen outputs a secret key sk from some key space $\mathcal{K}$,

$$sk \leftarrow \mathsf{KGen}$$

- Enc takes as input a secret key sk and a message m from some message space $\mathcal{M}$ and outputs a ciphertext c from some ciphertext space $\mathcal{C}$,

$$c = \mathsf{Enc}(sk, m)$$

- Dec takes as input a secret key sk and a ciphertext c and outputs a message m.

$$m = \mathsf{Dec}(sk, c)$$

Secret key encryption

We will return to this ad nauseam, but we just described a secret key encryption scheme (SKE). This is a triple of algorithms $\mathsf{SKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ such that

- KGen outputs a secret key sk from some key space $\mathcal{K}$,

$$sk \leftarrow \mathsf{KGen}$$

- Enc takes as input a secret key sk and a message m from some message space $\mathcal{M}$ and outputs a ciphertext c from some ciphertext space $\mathcal{C}$,

$$c = \mathsf{Enc}(sk, m)$$

- Dec takes as input a secret key sk and a ciphertext c and outputs a message m.

$$m = \mathsf{Dec}(sk, c)$$

We say nothing about correctness or security in the above, just syntax.

A thought experiment

Two lords, Francis and Henry, in their pavilions want to communicate 'securely'. In person, they decide between themselves

- an encryption method that uses a *secret key* to transform *message* into *ciphertext*,
- a decryption method that uses the same *secret key* to transform *ciphertext* into its underlying *message*,
- a *secret key* (they whisper it to each other to make sure no one hears).

When Francis wants to communicate a *message* with Henry he

- encrypts as above to create a *ciphertext* and writes it onto parchment,
- gives it to his most trusted rider to deliver to Henry,
- Henry then decrypts as above and reads the *message*.

What can go wrong?

Think. . .

What if the lords are not very clever and their encryption scheme is bad? What might ‘bad’ mean here?

What about the parchment and its *ciphertext*? Can it be altered, amended, lengthened, shortened, reordered (and still decrypt ‘meaningfully’)?

What about the rider? Can they be captured, killed, bribed, threatened, or forced to deliver late?

Basic functionality: correctness

We consider something fundamental relating to basic functionality.
Namely: correctness. (This is what you think it is.)

Basic functionality: correctness

We consider something fundamental relating to basic functionality.
Namely: correctness. (This is what you think it is.)

If:

a *secret key* is used to encrypt a *message* into a *ciphertext*,
i.e. $c = \mathrm{Enc}(sk, m)$,

then:

decrypting this *ciphertext* with the same *secret key* returns the same *message*,
i.e. $m = \mathrm{Dec}(sk, c)$.

Basic functionality: correctness

We consider something fundamental relating to basic functionality.
Namely: correctness. (This is what you think it is.)

If:

a *secret key* is used to encrypt a *message* into a *ciphertext*,
i.e. $c = \mathrm{Enc}(sk, m)$,

then:

decrypting this *ciphertext* with the same *secret key* returns the same *message*,
i.e. $m = \mathrm{Dec}(sk, c)$.

This says *nothing* about security!

A ‘good’ encryption scheme?

i. you cannot recover the *secret key* by observing *ciphertext(s)*.

Strong enough?

A ‘good’ encryption scheme?

i. you cannot recover the *secret key* by observing *ciphertext(s)*.

Strong enough?

No: consider the (silly) ‘do nothing’ encryption scheme that ignores the *secret key* and writes the *message* as the *ciphertext*.

A ‘good’ encryption scheme?

ii. you cannot recover *messages(s)* by observing *ciphertext(s)*.

Strong enough?

A ‘good’ encryption scheme?

ii. you cannot recover *messages(s)* by observing *ciphertext(s)*.

Strong enough?

No: imagine being able to recover only the first letter or word of a *message*.

A ‘good’ encryption scheme?

iii. you cannot recover even partial *message(s)* by observing *ciphertext(s)*.

Strong enough?

A ‘good’ encryption scheme?

iii. you cannot recover even partial *message(s)* by observing *ciphertext(s)*.

Strong enough?

No: imagine being able to recover statistical information about a *message* such as the number of times a given letter appears.

A ‘good’ encryption scheme?

iv. you cannot infer further ‘meaningful’ or ‘new’ information about *messages(s)* by observing *ciphertext(s)*.

Strong enough?

[2]The interested party should read Sections 1 and 2 of KL-IMC.

A ‘good’ encryption scheme?

iv. you cannot infer further ‘meaningful’ or ‘new’ information about *messages(s)* by observing *ciphertext(s)*.

Strong enough?

Getting warmer: this is not a formal statement,[2] but hopefully not being able to ‘infer’ or ‘learn’ information about underlying messages is intuited by the previous iterations.

What if the adversary has more abilities?

[2] The interested party should read Sections 1 and 2 of KL-IMC.

A ‘good’ encryption scheme?

v. you cannot infer further ‘meaningful’ or ‘new’ information about *messages(s)* by observing *ciphertext(s)* even given the ability to encrypt and decrypt selected *messages(s)* and *ciphertext(s)* of your choice.

Strong enough?

[3] The interested party should read Sections 3.2 and 3.4 of KL-IMC.

A ‘good’ encryption scheme?

v. you cannot infer further ‘meaningful’ or ‘new’ information about *messages(s)* by observing *ciphertext(s)* even given the ability to encrypt and decrypt selected *messages(s)* and *ciphertext(s)* of your choice.

Strong enough?

Yes*: this captures adversaries we expect to see. Is this realistic?[3]

[3] The interested party should read Sections 3.2 and 3.4 of KL-IMC.

Defining security is subtle!

In the previous exercise we implicitly considered two facets of what a definition should capture.

Defining security is subtle!

In the previous exercise we implicitly considered two facets of what a definition should capture.

What should not be possible:

- inferring further 'meaningful' information about *message(s)*.

Defining security is subtle!

In the previous exercise we implicitly considered two facets of what a definition should capture.

What should not be possible:

- inferring further 'meaningful' information about *message(s)*.

Given what capabilities:

- observing *ciphertext(s)*, or additionally
- the ability to encrypt and decrypt selected *message(s)* and *ciphertext(s)*.

Defining security is subtle!

In the previous exercise we implicitly considered two facets of what a definition should capture.

What should not be possible:

- inferring further ‘meaningful’ information about *message(s)*.

Given what capabilities:

- observing *ciphertext(s)*, or additionally
- the ability to encrypt and decrypt selected *message(s)* and *ciphertext(s)*.

Sometimes the catch all term of ‘confidentiality’ is used for this type of security. Some sources prefer ‘secrecy’.

Security beyond secrecy

Are there security properties we may want beyond secrecy?

Security beyond secrecy

Are there security properties we may want beyond secrecy?

What if the rider is bribed to alter the *ciphertext*? They may not know either the *secret key* or the *message*, but they can still reorder, add or remove characters.

Security beyond secrecy

Are there security properties we may want beyond secrecy?

What if the rider is bribed to alter the *ciphertext*? They may not know either the *secret key* or the *message*, but they can still reorder, add or remove characters.

What happens when Henry decrypts to receive the *message*, will it be gibberish? What if it is not?

Security beyond secrecy

Are there security properties we may want beyond secrecy?

What if the rider is bribed to alter the *ciphertext*? They may not know either the *secret key* or the *message*, but they can still reorder, add or remove characters.

What happens when Henry decrypts to receive the *message*, will it be gibberish? What if it is not?

More generally, assurance that the *message* or *ciphertext* received is equal to that which was sent is a separate and new desirable security property: 'integrity'.

Achieving integrity

Secrecy and integrity are distinct security properties, and are achieved via distinct means!

[4] The interested party should read Sections 4.1 and 4.2 of KL-IMC.

Achieving integrity

Secrecy and integrity are distinct security properties, and are achieved via distinct means! How might Francis go about ensuring integrity of the ciphertext?

[4] The interested party should read Sections 4.1 and 4.2 of KL-IMC.

Achieving integrity

Secrecy and integrity are distinct security properties, and are achieved via distinct means! How might Francis go about ensuring integrity of the ciphertext?

- ink that cannot be erased, filling all empty space on the parchment, 'tamper resistant' parchment etc (engineering approach),

[4] The interested party should read Sections 4.1 and 4.2 of KL-IMC.

Achieving integrity

Secrecy and integrity are distinct security properties, and are achieved via distinct means! How might Francis go about ensuring integrity of the ciphertext?

- ink that cannot be erased, filling all empty space on the parchment, 'tamper resistant' parchment etc (engineering approach),
- create a cryptographic *tag* that depends on the *ciphertext* and another shared secret *integrity key*,
- write the *ciphertext* and the *tag* on the parchment,
- Henry first checks the *tag* is correct using the *integrity key* before decrypting (cryptographic approach).[4]

[4] The interested party should read Sections 4.1 and 4.2 of KL-IMC.

Message authentication code

We just described a message authentication code (MAC). This is a triple of algorithms $\mathsf{MAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$ such that

- KGen outputs an integrity key ik from some key space $\mathcal{K}$,

$$ik \leftarrow \mathsf{KGen}$$

- Tag takes as input the integrity key ik and a message m from some message space $\mathcal{M}$ and outputs a tag τ from some tag space $\mathcal{T}$,

$$\tau = \mathsf{Tag}(ik, m)$$

- Vf takes as input the integrity key ik, a message m and a tag τ and outputs either 0 or 1.

$$\text{e.g. } 1 = \mathsf{Vf}(ik, m, \tau)$$

Message authentication code

We just described a message authentication code (MAC). This is a triple of algorithms $\mathsf{MAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$ such that

- KGen outputs an integrity key ik from some key space $\mathcal{K}$,

$$ik \leftarrow \mathsf{KGen}$$

- Tag takes as input the integrity key ik and a message m from some message space $\mathcal{M}$ and outputs a tag τ from some tag space $\mathcal{T}$,

$$\tau = \mathsf{Tag}(ik, m)$$

- Vf takes as input the integrity key ik, a message m and a tag τ and outputs either 0 or 1.

$$\text{e.g. } 1 = \mathsf{Vf}(ik, m, \tau)$$

Once again, nothing about correctness or security, only syntax.

Security beyond secrecy and integrity (!)

Are there yet more security properties we might desire?

Security beyond secrecy and integrity (!)

Are there yet more security properties we might desire?

Generally considered in the realm of cryptography

- preventing reordering *ciphertexts* and *tags*,
- preventing repeated *ciphertexts* and *tags*,
- preventing changing the direction of delivery of *ciphertexts* and *tags*.

Security beyond secrecy and integrity (!)

Are there yet more security properties we might desire?

Generally considered in the realm of cryptography

- preventing reordering *ciphertexts* and *tags*,
- preventing repeated *ciphertexts* and *tags*,
- preventing changing the direction of delivery of *ciphertexts* and *tags*.

Reaching into network engineering and security

- preventing delaying delivery of *ciphertexts* and *tags*,
- preventing denial of delivery of *ciphertexts* and *tags*.

From medieval metaphor to the modern day

Think...

From medieval metaphor to the modern day

Think. . . In the modern day, what are some reasonable analogies for

- Francis and Henry?
- the parchment?
- the rider?
- attacks against insufficient secrecy properties?
- the requirement for integrity?
- reordering, repeating, reflecting, delay and denial attacks?

Intermission fun fact!

Cryptography was considered a munition in the US $\rightarrow$ illegal to export its source code!

[5] Ronald J. Stay, Cryptic Controversy: U.S. Government Restrictions on Cryptography Exports and the Plight of Philip Zimmermann, 13 Ga. St. U. L. Rev. (2012). Available at: https://readingroom.law.gsu.edu/gsulr/vol13/iss2/14

[6] Constitutional Law. Free Speech Clause. Sixth Circuit Classifies Computer Source Code as Protected Speech. Junger v. Daley, 209 F.3d 481 (6th Cir. 2000). Harvard Law Review 114, no. 6 (2001): 1813–18. https://doi.org/10.2307/1342657

Intermission fun fact!

Cryptography was considered a munition in the US $\rightarrow$ illegal to export its source code!

Early versions of the NETSCAPE internet browser came in two flavours:

- the U.S. VERSION, with full length cryptographic keys, and
- the INTERNATIONAL VERSION, with weak cryptographic keys.

[5] Ronald J. Stay, Cryptic Controversy: U.S. Government Restrictions on Cryptography Exports and the Plight of Philip Zimmermann, 13 Ga. St. U. L. Rev. (2012). Available at: https://readingroom.law.gsu.edu/gsulr/vol13/iss2/14

[6] Constitutional Law. Free Speech Clause. Sixth Circuit Classifies Computer Source Code as Protected Speech. Junger v. Daley, 209 F.3d 481 (6th Cir. 2000). Harvard Law Review 114, no. 6 (2001): 1813–18. https://doi.org/10.2307/1342657

Intermission fun fact!

Cryptography was considered a munition in the US $\rightarrow$ illegal to export its source code!

Early versions of the NETSCAPE internet browser came in two flavours:

- the U.S. VERSION, with full length cryptographic keys, and
- the INTERNATIONAL VERSION, with weak cryptographic keys.

The creator of PGP email encryption was arrested after returning from Europe with its source code.[5]

[5] Ronald J. Stay, Cryptic Controversy: U.S. Government Restrictions on Cryptography Exports and the Plight of Philip Zimmermann, 13 Ga. St. U. L. Rev. (2012). Available at: https://readingroom.law.gsu.edu/gsulr/vol13/iss2/14

[6] Constitutional Law. Free Speech Clause. Sixth Circuit Classifies Computer Source Code as Protected Speech. Junger v. Daley, 209 F.3d 481 (6th Cir. 2000). Harvard Law Review 114, no. 6 (2001): 1813–18. https://doi.org/10.2307/1342657

Intermission fun fact!

Cryptography was considered a munition in the US $\rightarrow$ illegal to export its source code!

Early versions of the NETSCAPE internet browser came in two flavours:

- the U.S. VERSION, with full length cryptographic keys, and
- the INTERNATIONAL VERSION, with weak cryptographic keys.

The creator of PGP email encryption was arrested after returning from Europe with its source code.[5] The legal arguments ultimately involved the first amendment.[6]

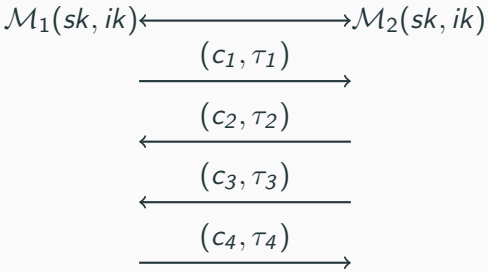
[5] Ronald J. Stay, Cryptic Controversy: U.S. Government Restrictions on Cryptography Exports and the Plight of Philip Zimmermann, 13 Ga. St. U. L. Rev. (2012). Available at: https://readingroom.law.gsu.edu/gsulr/vol13/iss2/14

[6] Constitutional Law. Free Speech Clause. Sixth Circuit Classifies Computer Source Code as Protected Speech. Junger v. Daley, 209 F.3d 481 (6th Cir. 2000). Harvard Law Review 114, no. 6 (2001): 1813–18. https://doi.org/10.2307/1342657

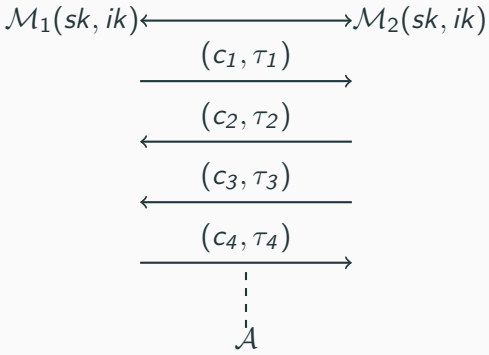
Machines (players, agents, principles, entities. . .)

$$\mathcal{M}_1(sk, ik) \longleftrightarrow \mathcal{M}_2(sk, ik)$$

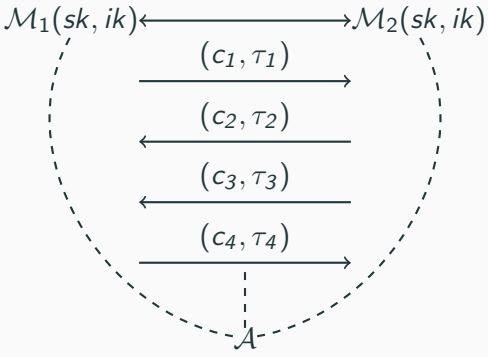
Machines (players, agents, principles, entities. . .)



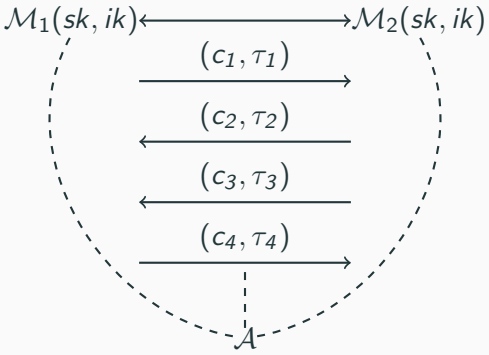
Machines (players, agents, principles, entities. . .)



Machines (players, agents, principles, entities. . .)

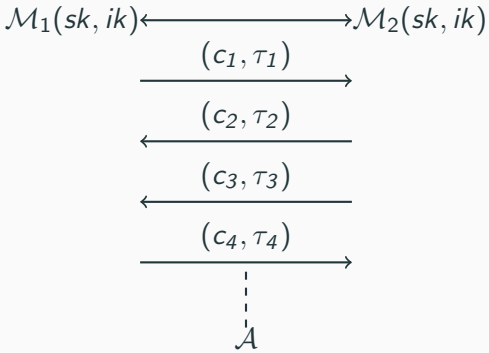


Machines (players, agents, principles, entities. . .)

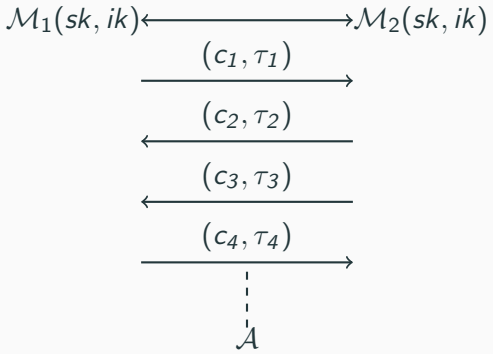


Often $(\mathcal{M}_1, \mathcal{M}_2)$ are called 'Alice' and 'Bob' and $\mathcal{A}$ is an adversary: 'Eve' the eavesdropper or 'Charlie' a more active adversary etc.

Generic security definitions

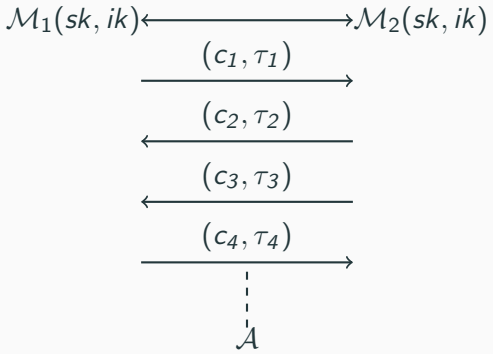


Generic security definitions



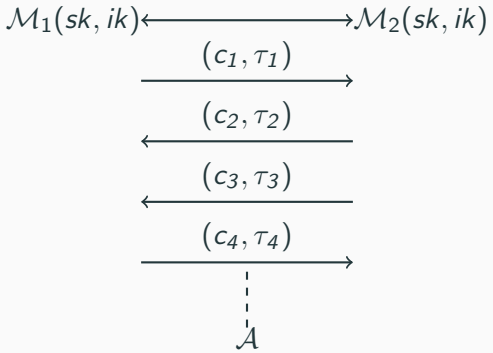
'Eve cannot infer anything meaningful about the underlying *messages*.'

Generic security definitions



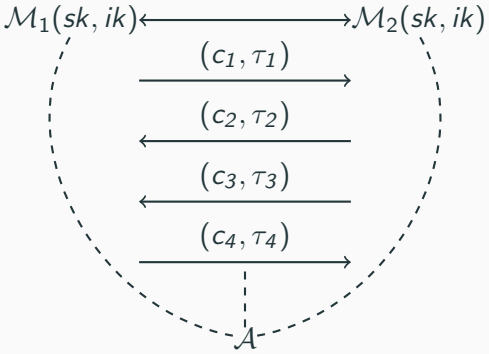
'Eve cannot infer anything meaningful about the underlying *messages*.' If you are certain an adversary can only eavesdrop, this security property is *generically* sufficient. Whether it is lords communicating between pavilions, or you talking to your friends, your bank or a journalist.

Generic security definitions

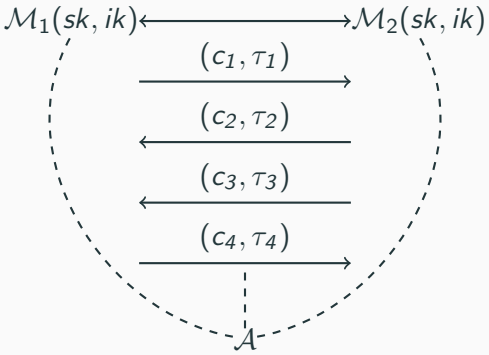


'Eve cannot infer anything meaningful about the underlying *messages*.' If you are certain an adversary can only eavesdrop, this security property is *generically* sufficient. Whether it is lords communicating between pavilions, or you talking to your friends, your bank or a journalist. (Eve is *not* usually considered a strong enough adversary.)

Generic security definitions



Generic security definitions



'Charlie can have selected *messages* encrypted and *ciphertexts* decrypted, but cannot infer anything meaningful about the underlying *messages* between Alice and Bob.'

The use of security definitions

This framework for thinking about security has much to recommend it. It

- explicitly gives an adversary capabilities and describes what they can or cannot achieve,
- allows a shared understanding of *ideas*, rather than nebulous (and varied!) understanding of English language semantics,
- allows us to be pessimistic about adversaries (i.e. assume they are stronger than they often will be) and therefore applies generically.

First lecture and tutorial

Please come with your questions! If you have them in advance, also write them on KEATS.

Topics for discussion may include (but are not limited to):

- formalisms of some of the security properties,
- 'reasonable' capabilities to allow adversaries,
- cryptography beyond secure *communications*,
- concrete and real world settings,
- *your suggestion here!*

The first problem sheet will concentrate on secrecy and integrity in different settings.