Security Notions for Bidirectional Channels

Giorgia Azzurra Marson  Bertram Poettering

FSE 2017
Tokyo, Japan
Outline

Secure channels and how they are modeled

Security notions for bidirectional channels

Analysis of bidirectional channel design
Communication channels

- setting: two-party communication over the Internet
- goal: deliver messages and preserve sending order
- how to achieve this: TCP/IP

Good, if there are only Alice and Bob (idealized world)
Cryptographic channels (a.k.a. secure channels)

- setting: two-party communication over the Internet
- goal: protect communication from adversaries

\[ m_1, m_2, m_3 \]
Cryptographic channels (a.k.a. secure channels)

- setting: two-party communication over the Internet
- goal: protect communication from adversaries
- security (informally): prevent eavesdropping
Cryptographic channels (a.k.a. secure channels)

- setting: two-party communication over the Internet
- goal: protect communication from adversaries
- security (informally): prevent eavesdropping and manipulation
Cryptographic channels (a.k.a. secure channels)

- setting: two-party communication over the Internet
- goal: **protect** communication from adversaries
- security (informally): prevent eavesdropping and manipulation
Cryptographic channels (a.k.a. secure channels)

- setting: two-party communication over the Internet
- goal: protect communication from adversaries
- security (informally): prevent eavesdropping and manipulation

make real world close to idealized world

I shall wait...

wait! · do not · buy now

do not · wait! · buy now

\[ m_1, m_2, m_3 \]

network

\[ m_2, m_1, m_3 \]
Modeling channel security [BKN’02]

Confidentiality

- intuitively: ciphertext hides plaintext
- formally: **IND-CPA** (a.k.a. ‘passive’)

\[ m^b \rightarrow c^* \rightarrow (m^0, m^1) \rightarrow b? \]
Modeling channel security [BKN’02]

Confidentiality

- intuitively: ciphertext hides plaintext
- formally: **IND-CPA** (a.k.a. ‘passive’) and **IND-CCA** (a.k.a. ‘active’)
Modeling channel security [BKN’02]

Confidentiality

- intuitively: ciphertext hides plaintext
- formally: IND-CPA (a.k.a. ‘passive’) and IND-CCA (a.k.a. ‘active’)

Integrity

- intuitively: manipulations are detected
- formally: INT-PTXT
Modeling channel security [BKN’02]

Confidentiality

• intuitively: ciphertext hides plaintext
• formally: IND-CPA (a.k.a. ‘passive’) and IND-CCA (a.k.a. ‘active’)

Integrity

• intuitively: manipulations are detected
• formally: INT-PTXT and INT-CTXT
Modeling channel security [BKN’02]

Confidentiality

- intuitively: ciphertext hides plaintext
- formally: **IND-CPA** (a.k.a. ‘passive’) and **IND-CCA** (a.k.a. ‘active’)

Integrity

- intuitively: manipulations are detected
- formally: **INT-PTXT** and **INT-CTXT**

Both incorporate replay and reordering protection
Cryptographic channels in theory: state of the art

- channel security: IND-CPA + **INT-CTXT** ($\iff$ **IND-CCA**)
- also called ‘stateful authenticated encryption’ (stateful AE)
- introduced to analyze (and prove) SSH channel security [BKN02]
- reference model to analyse TLS [JKSS12, KPW13, …]
Cryptographic channels in theory: state of the art

- channel security: IND-CPA + \textbf{INT-CTXT} \rightarrow IND-CCA
- also called ‘stateful authenticated encryption’ (stateful AE)
- introduced to analyze (and prove) SSH channel security [BKN02]
- reference model to analyse TLS [JKSS12,KPW13,\ldots]

stateful AE considered good abstraction of a secure channel
Channels are used for bidirectional communication

- prior work: ‘Sender → Receiver’ communication
- practice: channels protect bidirectional communication
- standard approach employs two independent unidirectional channels

**canonic composition of unidirectional channels**
Channels are used for bidirectional communication

- prior work: ‘Sender $\rightarrow$ Receiver’ communication
- practice: channels protect bidirectional communication
- standard approach employs two independent unidirectional channels
- does this yield a secure bidirectional channel?
- folklore: unidirectional security $\Rightarrow$ bidirectional security

**canonic composition of unidirectional channels**
Channels are used for bidirectional communication

- prior work: ‘Sender → Receiver’ communication
- practice: channels protect bidirectional communication
- standard approach employs two independent unidirectional channels
- does this yield a secure bidirectional channel?
- folklore: unidirectional security $\Rightarrow$ bidirectional security

What does it mean ‘bidirectional security’?

What is reordering?

What is an active attack?
Our contribution in a nutshell

Defining bidirectional security

- confidentiality: IND-2-CPA, IND-2-CCA
- integrity: INT-2-PTXT, INT-2-CTXT
- notions reflect that $\rightarrow$ and $\leftarrow$ are not independent of each other
Our contribution in a nutshell

Defining bidirectional security
- confidentiality: IND-2-CPA, IND-2-CCA
- integrity: INT-2-PTXT, INT-2-CTXT
- notions reflect that $\rightarrow$ and $\leftarrow$ are not independent of each other

Relations among notions
- \( \text{INT-2-CTXT} \implies \text{INT-2-PTXT} \)
- \( \text{IND-2-CCA} \implies \text{IND-2-CPA} \)
- \( \text{INT-2-CTXT} + \text{IND-2-CPA} \implies \text{IND-2-CCA} \)
Our contribution in a nutshell

Defining bidirectional security

- confidentiality: IND-2-CPA, IND-2-CCA
- integrity: INT-2-PTXT, INT-2-CTXT
- notions reflect that → and ← are not independent of each other

Relations among notions

- INT-2-CTXT $\implies$ INT-2-PTXT
- IND-2-CCA $\implies$ IND-2-CPA
- INT-2-CTXT + IND-2-CPA $\implies$ IND-2-CCA

Analysis of the canonic composition

- question: can security be lifted from unidirectional components?
- our results question common belief…
Active attacks in a bidirectional setting

active ≈ deviation from honest behavior

Manipulation of ciphertexts or of their order (akin to unidirectional setting)
Active attacks in a bidirectional setting
active \approx \text{deviation from honest behavior}

Manipulation of ciphertexts or of their order (akin to unidirectional setting)
Active attacks in a bidirectional setting

active \approx \text{deviation from honest behavior}

manipulation of ciphertexts or of their order (akin to unidirectional setting)
Active attacks in a bidirectional setting

active \approx \text{deviation from honest behavior}

Our model additionally allows to express that:

- ‘passive’ query may chronologically follow ‘active’ query (concurrency)

Manipulation of ciphertexts or of their order (akin to unidirectional setting)
Active attacks in a bidirectional setting

active \approx \text{deviation from honest behavior}

Manipulation of ciphertexts or of their order (akin to unidirectional setting)

Our model additionally allows to express that:

- ‘passive’ query may chronologically follow ‘active’ query (concurrency)
Active attacks in a bidirectional setting

active $\approx$ deviation from honest behavior

Our model additionally allows to express that:

- ‘passive’ query may chronologically follow ‘active’ query (concurrency)
- active attack on $\leftarrow$ may influence security of $\rightarrow$

Entities:
- $s$ for sender
- $r$ for receiver

Messages:
- $c_1$ from $s$ to $r$
- $c_2$ from $r$ to $s$
- $c'$ from $r$ to $s$
Bidirectional security of the canonic composition

Generic analysis: can security be lifted from unidirectional components?

- INT-PTXT + INT-PTXT $\implies$ INT-2-PTXT
- INT-CTXT + INT-CTXT $\implies$ INT-2-CTXT
- IND-CPA + IND-CPA $\implies$ INT-2-CPA

Bidirectional security of TLS and SSH (the good news)

- TLS and SSH channel offer stateful AE security [K01,BKN02,PRS11]
- Encode-then-E&M for SSH, CBC-based M-then-E for TLS
- our result: they also offer IND-2-CCA and INT-2-CTXT security
Bidirectional security of the canonic composition

Generic analysis: can security be lifted from unidirectional components?

- \( \text{INT-PTXT} + \text{INT-PTXT} \Rightarrow \text{INT-2-PTXT} \)
- \( \text{INT-CTXT} + \text{INT-CTXT} \Rightarrow \text{INT-2-CTXT} \)
- \( \text{IND-CPA} + \text{IND-CPA} \Rightarrow \text{INT-2-CPA} \)
- \( \text{IND-CCA} + \text{IND-CCA} \not\Rightarrow \text{INT-2-CCA} \)

Bidirectional security of TLS and SSH (the good news)

- TLS and SSH channel offer stateful AE security \([K01,BKN02,PRS11]\)
- Our result: they also offer \(\text{IND-2-CCA} \) and \(\text{INT-2-CTXT} \) security
Bidirectional security of the canonic composition

Generic analysis: can security be lifted from unidirectional components?

- $\text{INT-PTXT} + \text{INT-PTXT} \implies \text{INT-2-PTXT}$
- $\text{INT-CTXT} + \text{INT-CTXT} \implies \text{INT-2-CTXT}$
- $\text{IND-CPA} + \text{IND-CPA} \implies \text{INT-2-CPA}$
- $\text{IND-CCA} + \text{IND-CCA} \not\equiv \text{INT-2-CCA}$

Bidirectional security of TLS and SSH (the good news)

- TLS and SSH channel offer stateful AE security $[K01,BKN02,PRS11]$  
  Encode-then-E&M for SSH, CBC-based M-then-E for TLS
- our result: they also offer $\text{IND-2-CCA}$ and $\text{INT-2-CTXT}$ security
Summary

This work

- formalize security notions for bidirectional channels
- analyze ‘canonic composition’
- confirm security of (crypto core of) TLS and SSH channels
Summary

This work

- formalize security notions for bidirectional channels
- analyze ‘canonic composition’
- confirm security of (crypto core of) TLS and SSH channels

Future work & open questions

- channel security in a multi-party setting (work in progress)
- bidirectional security of real TLS and SSH (beyond crypto core)
Summary

This work

- formalize security notions for bidirectional channels
- analyze ‘canonic composition’
- confirm security of (crypto core of) TLS and SSH channels

Future work & open questions

- channel security in a multi-party setting (work in progress)
- bidirectional security of real TLS and SSH (beyond crypto core)

Thank you!
Defining bidirectional confidentiality (IND-2-CCA)

**Send** \( (u, m^0, m^1) \)

\[ c^* \leftarrow \text{Send}(st_u, m^b) \]

if \( h_u = \text{True} \)

\[ C_u[s_u] \leftarrow c^* \]

\[ s_u \leftarrow s_u + 1 \]

Return \( c^* \)

**Recv** \( (u, c) \)

\[ m \leftarrow \text{Recv}(st_u, c) \]

if \( r_u < s_v \) and \( c = C_v[r_u] \)

\[ r_u \leftarrow r_u + 1 \]

else

\[ h_u \leftarrow \text{False} \]

Return \( h_u? \diamond : m \)
Defining bidirectional confidentiality (IND-2-CCA)

Send \((u, m^0, m^1)\)

\[ c^* \leftarrow \text{Send}(st_u, m^b) \]

if \(h_u = \text{True}\)

\[ C_u[s_u] \leftarrow c^* \]

\[ s_u \leftarrow s_u + 1 \]

Return \(c^*\)

Recv \((u, c)\)

\[ m \leftarrow \text{Recv}(st_u, c) \]

if \(r_u < s_v \text{ and } c = C_v[r_u]\)

\[ r_u \leftarrow r_u + 1 \]

else

\[ h_u \leftarrow \text{False} \]

Return \(h_u?\diamond : m\)
Defining bidirectional confidentiality (IND-2-CCA)

**Send** \((u, m^0, m^1)\)
- \(c^* \leftarrow \text{Send}(st_u, m^b)\)
- if \(h_u = \text{True}\)
  - \(C_u[s_u] \leftarrow c^*\)
  - \(s_u \leftarrow s_u + 1\)
- Return \(c^*\)

**Recv** \((u, c)\)
- \(m \leftarrow \text{Recv}(st_u, c)\)
- if \(r_u < s_v\) and \(c = C_v[r_u]\)
  - \(r_u \leftarrow r_u + 1\)
  - else
  - \(h_u \leftarrow \text{False}\)
- Return \(h_u?\) : \(m\)
Defining bidirectional confidentiality (IND-2-CCA)

**Send** \((u, m^0, m^1)\)

\[ c^* \leftarrow \text{Send}(st_u, m^b) \]

if \(h_u = \text{True}\)

\[ C_u[s_u] \leftarrow c^* \]

\[ s_u \leftarrow s_u + 1 \]

Return \(c^*\)

**Recv** \((u, c)\)

\[ m \leftarrow \text{Recv}(st_u, c) \]

if \(r_u < s_v\) and \(c = C_v[r_u]\)

\[ r_u \leftarrow r_u + 1 \]

else

\[ h_u \leftarrow \text{False} \]

Return \(h_u\)? \(\diamond : m\)
Defining bidirectional confidentiality (IND-2-CCA)

**Send** \((u, m^0, m^1)\)

\[c^* \leftarrow \text{Send}(st_u, m^b)\]

if \(h_u = \text{True}\)

\[C_u[s_u] \leftarrow c^*\]

\[s_u \leftarrow s_u + 1\]

Return \(c^*\)

**Recv** \((u, c)\)

\[m \leftarrow \text{Recv}(st_u, c)\]

if \(r_u < s_v\) and \(c = C_v[r_u]\)

\[r_u \leftarrow r_u + 1\]

else

\[h_u \leftarrow \text{False}\]

Return \(h_u? \odot : m\)
Defining bidirectional confidentiality (IND-2-CCA)

**Send** \((u, m^0, m^1)\)

\[c^* \leftarrow \text{Send}(st_u, m^b)\]

if \(h_u = \text{True}\)

\[C_u[s_u] \leftarrow c^*\]

\[s_u \leftarrow s_u + 1\]

Return \(c^*\)

**Recv** \((u, c)\)

\[m \leftarrow \text{Recv}(st_u, c)\]

if \(r_u < s_v \text{ and } c = C_v[r_u]\)

\[r_u \leftarrow r_u + 1\]

else

\[h_u \leftarrow \text{False}\]

Return \(h_u? \odot : m\)