

Cryptographic protocols

Myrto Arapinis
School of Informatics
University of Edinburgh

February 15, 2019

Context

Applications exchanging **sensitive data** over a **public network**:

- ▶ eBanking,
- ▶ eCommerce,
- ▶ eVoting,
- ▶ ePassports,
- ▶ Mobile phones,
- ▶ ...

Context

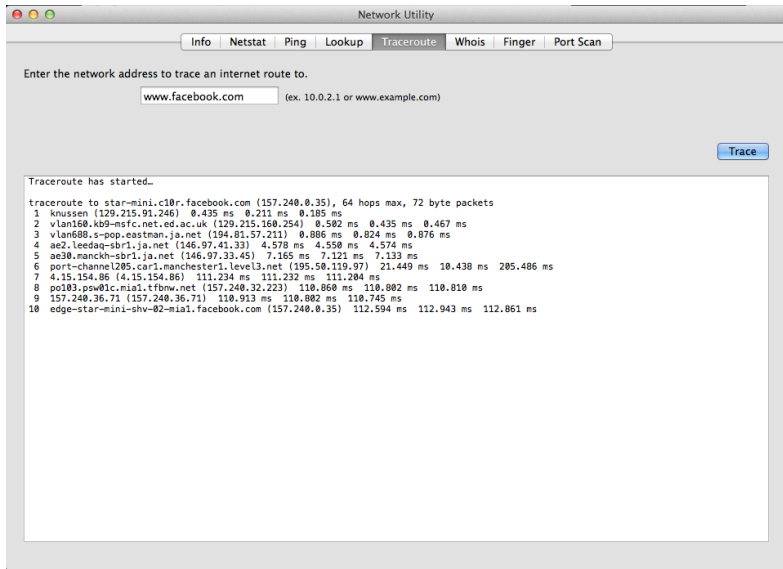
Applications exchanging **sensitive data** over a **public network**:

- ▶ eBanking,
- ▶ eCommerce,
- ▶ eVoting,
- ▶ ePassports,
- ▶ Mobile phones,
- ▶ ...

A malicious agent can:

- ▶ record, alter, delete, insert, redirect, reorder, and reuse past or current messages, and inject new messages
→ **the network is the attacker**
- ▶ control dishonest participants

The attacker controls the network (1)



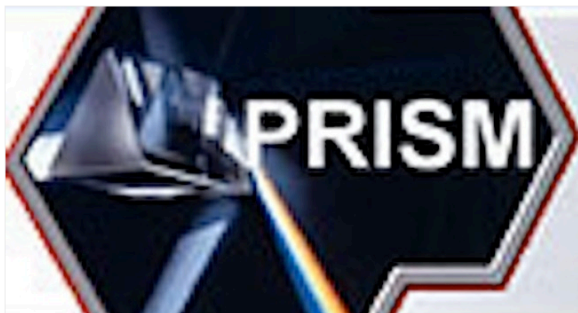
The attacker controls the network (2)



Networks

Verizon, BT, Vodafone, Level 3 'let NSA jack into Google, Yahoo! fiber'

Telcos cooperated with g-men in data slurp, claim sources

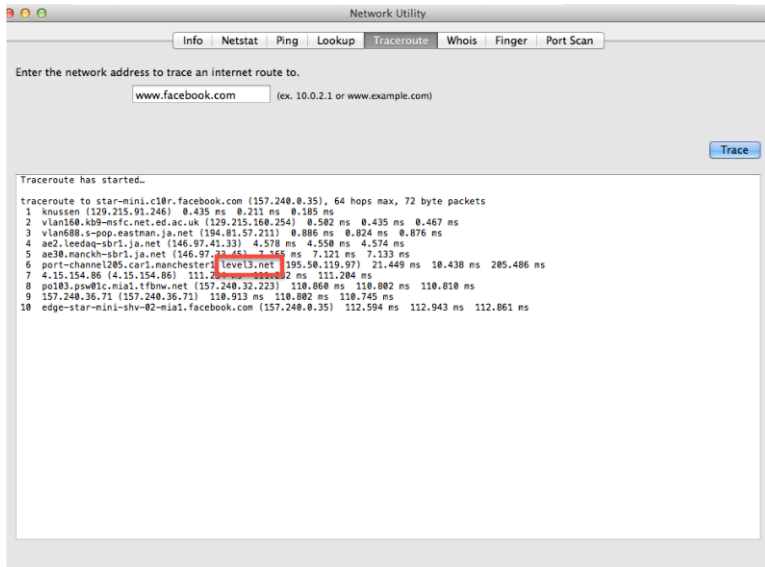


27 Nov 2013 at 02:19, [Shaun Nichols](#)



In October, NSA whistleblower Edward Snowden claimed Uncle Sam's spies [tapped into the optic-fiber cables](#) linking the data centers of Google and Yahoo!

The attacker controls the network (3)



Network Utility

Info Netstat Ping Lookup **Traceroute** Whois Finger Port Scan

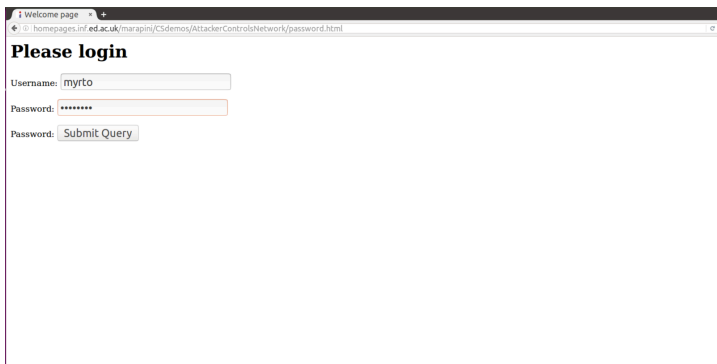
Enter the network address to trace an internet route to.

(ex. 10.0.2.1 or www.example.com)

Traceroute has started...

```
traceroute to star-mini.c10r.facebook.com (157.240.0.35), 64 hops max, 72 byte packets
 1 knussen (129.215.91.246)  0.435 ms  0.211 ms  0.185 ms
 2 vln160.kb9-msfc.net.ed.ac.uk (129.215.160.254)  0.502 ms  0.435 ms  0.467 ms
 3 vln688.s-pop.eastman.ja.net (194.81.57.211)  0.806 ms  0.824 ms  0.876 ms
 4 ae2.leedaq-sbr1.ja.net (146.97.41.33)  4.578 ms  4.550 ms  4.574 ms
 5 ae30.manckh-sbr1.ja.net (146.97.33.45)  7.165 ms  7.121 ms  7.133 ms
 6 port-channel205.car1.manchester1.level3.net (195.50.119.97)  21.449 ms  10.438 ms  205.486 ms
 7 4.15.154.86 (4.15.154.86)  111.204 ms  111.204 ms  111.204 ms
 8 poi03.psw@ic.mia1.tfbnw.net (157.240.32.223)  110.860 ms  110.802 ms  110.810 ms
 9 157.240.36.71 (157.240.36.71)  110.913 ms  110.802 ms  110.745 ms
10 edge-star-mini-shv-02-mia1.facebook.com (157.240.0.35)  112.594 ms  112.943 ms  112.861 ms
```

All messages can be intercepted by an attacker (1)



The screenshot shows a web browser window with a single tab titled "Welcome page". The address bar displays the URL "homepages.inf.ed.ac.uk/marapini/C5demos/AttackerControlsNetwork/password.html". The page content begins with the heading "Please login". Below this, there are three input fields: a "Username:" field containing the text "myrto", a "Password:" field filled with seven asterisks, and a "Submit Query" button. A vertical red line is visible on the left side of the browser window, likely indicating a security warning or a network-related message.

Welcome page

homepages.inf.ed.ac.uk/marapini/C5demos/AttackerControlsNetwork/password.html

Please login

Username: myrto

Password: *****

Password:

All messages can be intercepted by an attacker (2)

The image shows a Wireshark packet capture interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with icons for various functions. The main display area is divided into three panes: a packet list on the left, a packet details pane in the middle, and a packet bytes pane on the right.

The packet list pane shows a list of captured packets. The selected packet is packet 14, which is a POST request to `/warapini/C5demos/AttackerControlNetwork/password.html`. The details pane shows the structure of this packet, including the Ethernet II header, Internet Protocol Version 4 header, Transmission Control Protocol header, and Hypertext Transfer Protocol header. The packet bytes pane shows the raw data of the packet, with a red box highlighting the `...unary rto&w=12345678` string.

Packet 14 details:

- Ethernet II, Src: VMware, Se:00:0c:29:5e:00:02, Dst: VMware, F0:7d:7d:d2
- Internet Protocol Version 4, Src: 172.16.76.155, Dst: 129.215.32.13
- Transmission Control Protocol, Src Port: 36412 (36412), Dst Port: 80 (80), Seq: 636, Ack: 610, Len: 609
- Hypertext Transfer Protocol
- POST /warapini/C5demos/AttackerControlNetwork/password.html HTTP/1.1\r\n
- Host: homepages.inf.ed.ac.uk\r\n
- User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:49.0) Gecko/20100101 Firefox/49.0\r\n
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
- Accept-Language: en-us,en;q=0.8\r\n
- Accept-Encoding: gzip, deflate\r\n
- Referer: http://homepages.inf.ed.ac.uk/warapini/C5demos/AttackerControlNetwork/password.html\r\n
- Cookie: gn041.3.388514529.1476874824\r\n
- Connection: keep-alive\r\n
- Upgrade-Insecure-Requests: 1\r\n
- Content-Type: application/x-www-form-urlencoded\r\n
- Content-Length: 20\r\n
- \r\n
- [Full request URI: http://homepages.inf.ed.ac.uk/warapini/C5demos/AttackerControlNetwork/password.html]

Packet bytes pane highlights the string: `...unary rto&w=12345678`

All messages can be intercepted by an attacker (2)

The image shows a Wireshark packet capture interface. The top pane displays a list of network packets. The bottom pane shows the details of a selected packet (Frame 14: 663 bytes on wire).

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
10	0.000000000	172.16.76.155	172.16.76.2	DNS	82	Standard query 0xe31b A homepages.inf.ed.ac.uk
26	0.023612151	172.16.76.155	172.16.76.2	DNS	82	Standard query 0xd17a A homepages.inf.ed.ac.uk
36	0.073200501	172.16.76.2	172.16.76.155	DNS	453	Standard query response 0xe31b A homepages.inf.ed.ac.uk A 129.215.32.13 NS lewis.ucs.ed.ac.uk NS xlab-0.ed.ac.uk NS cancer.ucs.ed.ac.uk NS d.
46	0.074843379	172.16.76.2	172.16.76.155	DNS	453	Standard query response 0xd17a A homepages.inf.ed.ac.uk A 129.215.32.13 NS xlab-0.ed.ac.uk NS cancer.ucs.ed.ac.uk NS dns2.inf.ed.ac.uk NS d.
56	0.077897581	172.16.76.2	172.16.76.155	DNS	82	Standard query 0x4997 AAAA homepages.inf.ed.ac.uk
66	0.079077511	172.16.76.155	129.215.32.13	TCP	74	36412 - 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1200371 TSecr=0 WS=128
76	0.127399000	172.16.76.2	172.16.76.155	DNS	134	Standard query response 0x4997 AAAA homepages.inf.ed.ac.uk SOA dns2.inf.ed.ac.uk
86	0.160218320	129.215.32.13	172.16.76.155	TCP	60	80 - 36412 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
96	0.160287872	172.16.76.155	129.215.32.13	TCP	54	36412 - 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
106	0.160286984	172.16.76.155	129.215.32.13	HTTP	669	POST /warapini/Csdemos/AttackerControlNetwork/password.html HTTP/1.1 (application/x-www-form-urlencoded)
116	0.160826180	129.215.32.13	172.16.76.155	TCP	60	80 - 36412 [ACK] Seq=1 Ack=336 Win=64240 Len=0
126	0.234165479	129.215.32.13	172.16.76.155	HTTP	663	HTTP/1.1 200 OK (text/html)
136	0.234191620	172.16.76.155	129.215.32.13	TCP	54	36412 - 80 [ACK] Seq=636 Ack=610 Win=39841 Len=0
146	0.902511530	172.16.76.155	129.215.32.13	HTTP	663	POST /warapini/Csdemos/AttackerControlNetwork/password.html HTTP/1.1 (application/x-www-form-urlencoded)
156	0.904729220	129.215.32.13	172.16.76.155	TCP	60	80 - 36412 [ACK] Seq=610 Ack=1245 Win=64240 Len=0
166	0.145430844	129.215.32.13	172.16.76.155	HTTP	662	HTTP/1.1 200 OK (text/html)
176	0.145496842	172.16.76.155	129.215.32.13	TCP	54	36412 - 80 [ACK] Seq=1245 Ack=1218 Win=31059 Len=0

Packet Details (Frame 14):

- Frame 14: 663 bytes on wire (5304 bits), 663 bytes captured (5304 bits) on interface 0
- Ethernet II, Src: VMware, Src: 00:0c:29:5e:00:02, Dst: VMware, Dst: 00:50:56:f0:7d:d2
- Internet Protocol Version 4, Src: 172.16.76.155, Dst: 129.215.32.13
- Transmission Control Protocol, Src Port: 36412 (36412), Dst Port: 80 (80), Seq: 636, Ack: 610, Len: 669
- Hypertext Transfer Protocol
- POST /warapini/Csdemos/AttackerControlNetwork/password.html HTTP/1.1\r\n
- Host: homepages.inf.ed.ac.uk\r\n
- User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:49.0) Gecko/20100101 Firefox/49.0\r\n
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
- Accept-Language: en-US,en;q=0.8\r\n
- Accept-Encoding: gzip, deflate\r\n
- Referer: http://homepages.inf.ed.ac.uk/warapini/Csdemos/AttackerControlNetwork/password.html\r\n
- Cookie: gn0k1.3.388514529.1478874824\r\n
- Connection: keep-alive\r\n
- Upgrade-Insecure-Requests: 1\r\n
- Content-Type: application/x-www-form-urlencoded\r\n
- Content-Length: 20\r\n
- \r\n
- [Full request URI: http://homepages.inf.ed.ac.uk/warapini/Csdemos/AttackerControlNetwork/password.html]
- HTTP message: 1\r\n
- 3a 20 6b 65 65 70 2d 61 6c 69 76 65 6d 0a 55 70 : keep-a live..up
- 67 72 61 64 65 2d 49 6e 79 65 63 75 72 65 2d 52 grade-in secure-#
- 61 75 65 73 74 73 3a 2b 31 6d 0a 43 6f 6e 74 requests: 1, 1, cont
- 65 74 2d 54 79 70 65 3a 20 61 70 70 6c 69 63 ent-type: applic
- 61 74 69 6f 6e 2f 78 2d 77 77 2d 6b 6f 72 6d ation/x-ww-form
- 70 72 6c 65 6e 65 6f 64 65 64 0d 0a 43 6f 6e -urled cod ded..com
- 74 75 6e 74 2d 4c 65 6e 67 74 68 3a 20 32 30 9d
- 0d 0d 0a 75 6e 3d 6d 79 72 74 6f 2d 70 77 3d 31
- 32 34 35 36 37 38 ...unway rto&w=12345678

An attacker can **intercept** packets, but also **alter**, **forge new**, and **inject** packets

More complex systems needed...

More complex systems needed...



$$\begin{array}{c} e = E(K_E, \text{Transfer 100 € on Amazon's account}) \\ \hline m = \text{MAC}(K_M, E(K_E, \text{Transfer 100 € on Amazon's account})) \end{array} \rightarrow$$



More complex systems needed...



$$\begin{array}{c} e = E(K_E, \text{Transfer 100 € on Amazon's account}) \\ \hline m = \text{MAC}(K_M, E(K_E, \text{Transfer 100 € on Amazon's account})) \end{array} \rightarrow$$



Replay attack



$$\xrightarrow{(e, m)}$$



$$\xrightarrow{(e, m)}$$

\vdots

$$\xrightarrow{(e, m)}$$



... to achieve more complex properties

- ▶ **Confidentiality:** Some information should never be revealed to unauthorised entities.
- ▶ **Integrity:** Data should not be altered in an unauthorised manner since the time it was created, transmitted or stored by an authorised source.
- ▶ **Authentication:** Ability to know with certainty the identity of an communicating entity.
- ▶ **Anonymity:** The identity of the author of an action (e.g. sending a message) should not be revealed.
- ▶ **Unlinkability:** An attacker should not be able to deduce whether different services are delivered to the same user
- ▶ **Non-repudiation:** The author of an action should not be able to deny having triggered this action.
- ▶ ...

Cryptographic protocols

Cryptographic protocols

Programs relying on **cryptographic primitives** and whose goal is the establishment of “**secure**” communications.

Cryptographic protocols

Cryptographic protocols

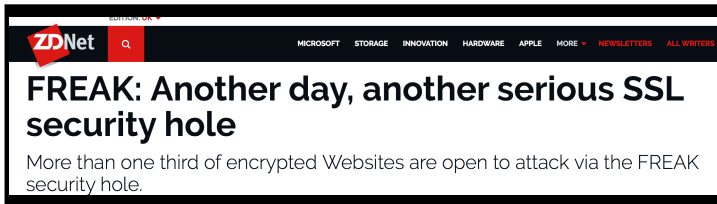
Programs relying on **cryptographic primitives** and whose goal is the establishment of “**secure**” **communications**.

But!

Many exploitable errors are due not to design errors in the primitives, but to the way they are used, *i.e.* bad protocol design and buggy or not careful enough implementation

Numerous deployed protocols are flawed...

... and end up in the news :(



EDITOR: UK

ZDNet

MICROSOFT STORAGE INNOVATION HARDWARE APPLE MORE NEWSLETTERS ALL WRITERS

FREAK: Another day, another serious SSL security hole

More than one third of encrypted Websites are open to attack via the FREAK security hole.



The Telegraph

Home Video News **World** Sport Business Money Comment Culture Travel Life W

USA Asia China Europe Middle East Australasia Africa South America Central Asia

HOME » NEWS » WORLD NEWS » NORTH AMERICA » USA

Hacker remotely crashes Jeep from 10 miles away

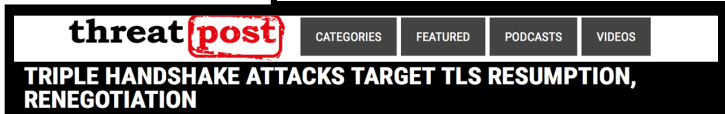
Security experts warn that more than 470,000 cars made by Fiat Chrysler could be at risk of being attacked by similar means – including those driven in the UK



The Register[®]
Biting the hand that feeds IT

Defects in e-passports allow real-time tracking

This threat brought to you by RFID



threat **post**

CATEGORIES FEATURED PODCASTS VIDEOS

TRIPLE HANDSHAKE ATTACKS TARGET TLS RESUMPTION, RENEGOTIATION

Logical attacks

Many of these attacks do not even break the crypto primitives!!

Example of a logical attack

Assume a commutative symmetric encryption scheme

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

where $\{m\}_k$ denotes the encryption of message m under the key k

Example: stream ciphers

Example of a logical attack

Assume a commutative symmetric encryption scheme

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

where $\{m\}_k$ denotes the encryption of message m under the key k

Example: stream ciphers

A

|

|

B

|

|

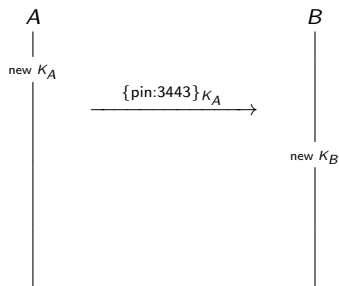
Example of a logical attack

Assume a commutative symmetric encryption scheme

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

where $\{m\}_k$ denotes the encryption of message m under the key k

Example: stream ciphers



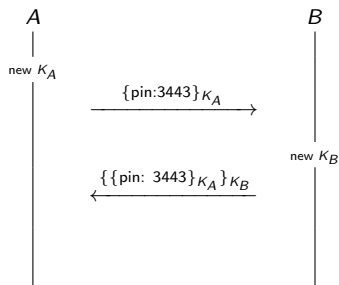
Example of a logical attack

Assume a commutative symmetric encryption scheme

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

where $\{m\}_k$ denotes the encryption of message m under the key k

Example: stream ciphers



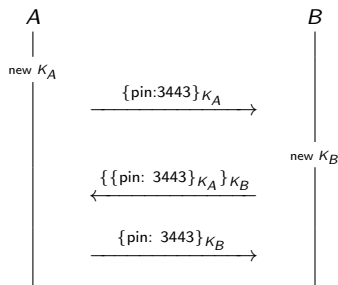
Example of a logical attack

Assume a commutative symmetric encryption scheme

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

where $\{m\}_k$ denotes the encryption of message m under the key k

Example: stream ciphers



since $\{\{\text{pin: 3443}\}_{K_A}\}_{K_B} = \{\{\text{pin: 3443}\}_{K_B}\}_{K_A}$ by commutativity

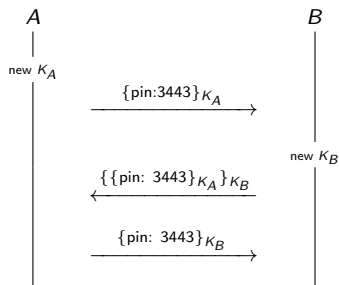
Example of a logical attack

Assume a commutative symmetric encryption scheme

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

where $\{m\}_k$ denotes the encryption of message m under the key k

Example: stream ciphers



No authentication!

since $\{\{\text{pin: 3443}\}_{K_A}\}_{K_B} = \{\{\text{pin: 3443}\}_{K_B}\}_{K_A}$ by commutativity

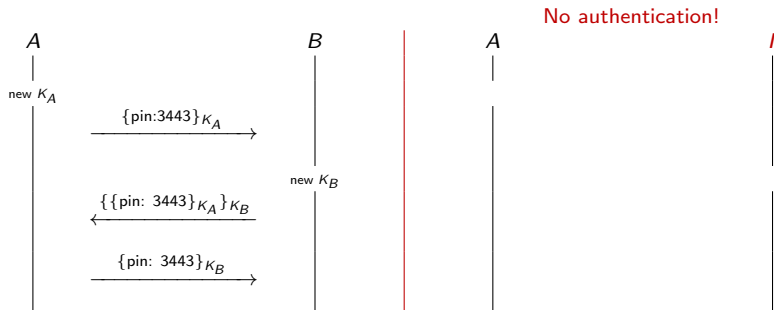
Example of a logical attack

Assume a commutative symmetric encryption scheme

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

where $\{m\}_k$ denotes the encryption of message m under the key k

Example: stream ciphers



since $\{\{\text{pin: 3443}\}_{K_A}\}_{K_B} = \{\{\text{pin: 3443}\}_{K_B}\}_{K_A}$ by commutativity

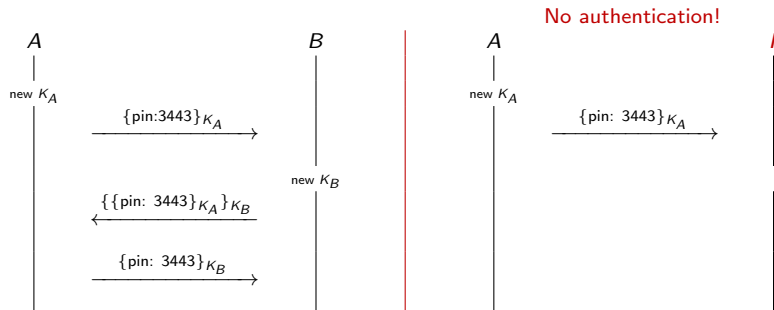
Example of a logical attack

Assume a commutative symmetric encryption scheme

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

where $\{m\}_k$ denotes the encryption of message m under the key k

Example: stream ciphers



since $\{\{\text{pin: 3443}\}_{K_A}\}_{K_B} = \{\{\text{pin: 3443}\}_{K_B}\}_{K_A}$ by commutativity

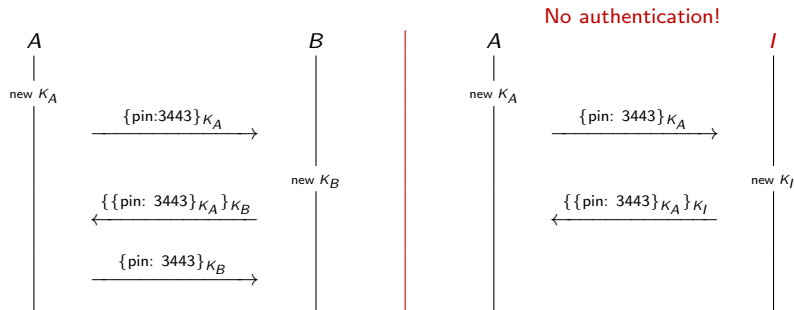
Example of a logical attack

Assume a commutative symmetric encryption scheme

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

where $\{m\}_k$ denotes the encryption of message m under the key k

Example: stream ciphers



since $\{\{\text{pin: 3443}\}_{K_A}\}_{K_B} = \{\{\text{pin: 3443}\}_{K_B}\}_{K_A}$ by commutativity

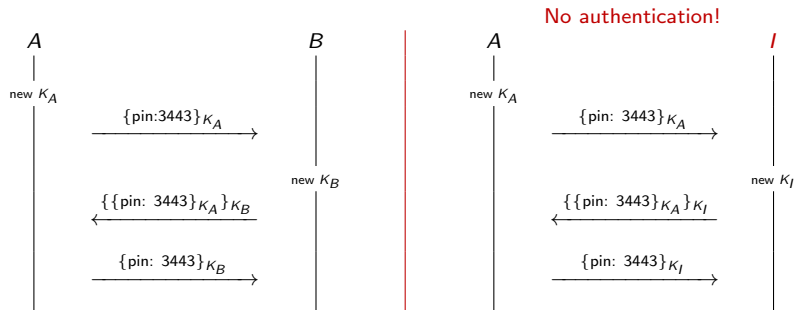
Example of a logical attack

Assume a commutative symmetric encryption scheme

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

where $\{m\}_k$ denotes the encryption of message m under the key k

Example: stream ciphers



since $\{\{\text{pin: 3443}\}_{K_A}\}_{K_B} = \{\{\text{pin: 3443}\}_{K_B}\}_{K_A}$ by commutativity

Authentication and key agreement protocols

Authentication and key agreement

- ▶ Long-term keys should be used as little as possible to reduce “attack-surface”
- ▶ The use of a key should be restricted to a specific purpose
e.g. you shouldn't use the same RSA key both for encryption and signing
- ▶ Public key algorithms tend to be computationally more expensive than symmetric key algorithms
- ↪ Long-term keys are used to establish short-term **session keys**
e.g. TLS over HTTP, AKA for 3G, BAC for epassports, etc.

Needham-Schroeder Public Key (NSPK)

NSPK: authentication and key agreement protocol

A

|

|

B

|

|

[N. Roger, M. Schroeder, Michael. "Using encryption for authentication in large networks of computers". Communications of the ACM (December 1978)]

Needham-Schroeder Public Key (NSPK)

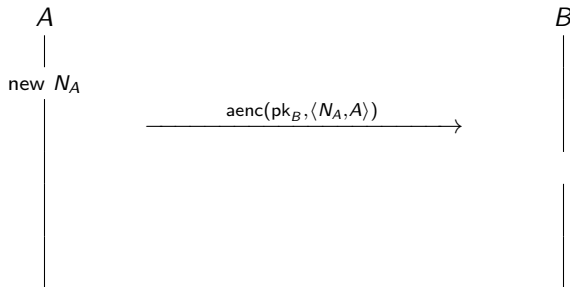
NSPK: authentication and key agreement protocol



[N. Roger, M. Schroeder, Michael. "Using encryption for authentication in large networks of computers". Communications of the ACM (December 1978)]

Needham-Schroeder Public Key (NSPK)

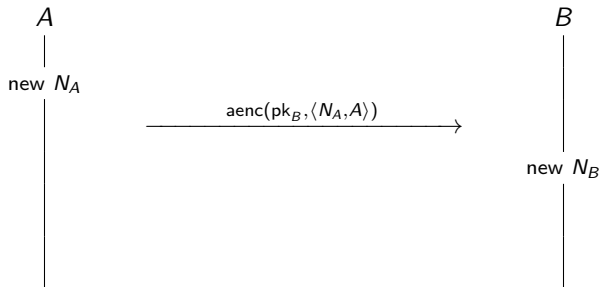
NSPK: authentication and key agreement protocol



[N. Roger, M. Schroeder, Michael. "Using encryption for authentication in large networks of computers". Communications of the ACM (December 1978)]

Needham-Schroeder Public Key (NSPK)

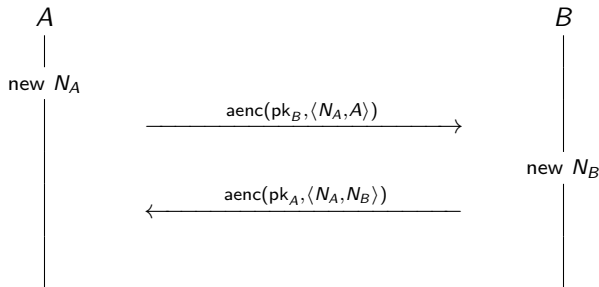
NSPK: authentication and key agreement protocol



[N. Roger, M. Schroeder, Michael. "Using encryption for authentication in large networks of computers". Communications of the ACM (December 1978)]

Needham-Schroeder Public Key (NSPK)

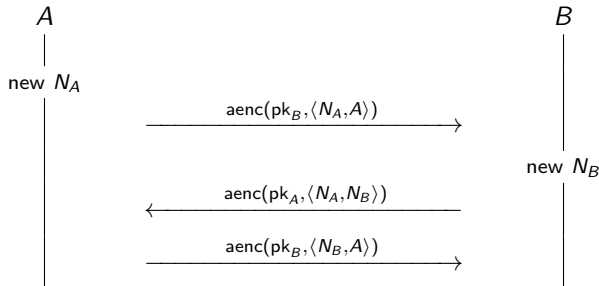
NSPK: authentication and key agreement protocol



[N. Roger, M. Schroeder, Michael. "Using encryption for authentication in large networks of computers". Communications of the ACM (December 1978)]

Needham-Schroeder Public Key (NSPK)

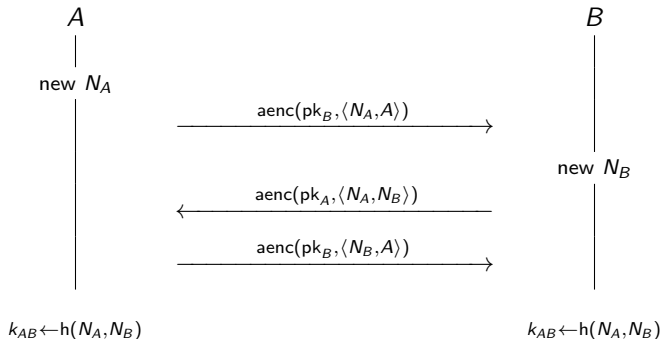
NSPK: authentication and key agreement protocol



[N. Roger, M. Schroeder, Michael. "Using encryption for authentication in large networks of computers". Communications of the ACM (December 1978)]

Needham-Schroeder Public Key (NSPK)

NSPK: authentication and key agreement protocol



[N. Roger, M. Schroeder, Michael. "Using encryption for authentication in large networks of computers". Communications of the ACM (December 1978)]

NSPK: security requirements

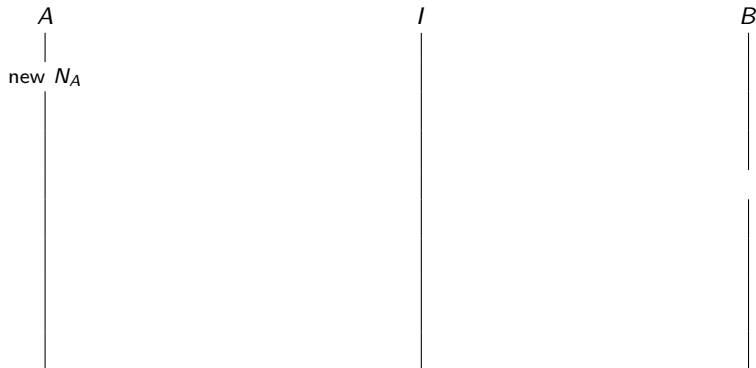
- ▶ **Authentication:** if Alice has completed the protocol, apparently with Bob, then Bob must also have completed the protocol with Alice.
- ▶ **Authentication:** If Bob has completed the protocol, apparently with Alice, then Alice must have completed the protocol with Bob.
- ▶ **Confidentiality:** Messages sent encrypted with the agreed key ($k \leftarrow h(N_A, NB)$) remain secret.

NSPK: Lowe's attack on authentication

Attack found 17 years after the publication of the NS protocol!!

NSPK: Lowe's attack on authentication

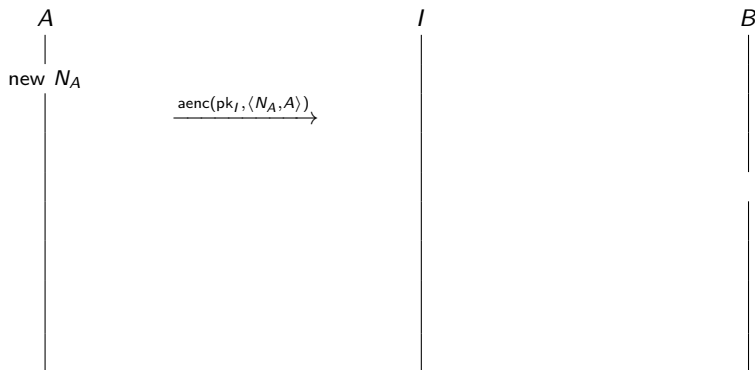
Attack found 17 years after the publication of the NS protocol!!



[G. Lowe. "An attack on the Needham-Schroeder public key authentication protocol". Information Processing Letters (November 1995)]

NSPK: Lowe's attack on authentication

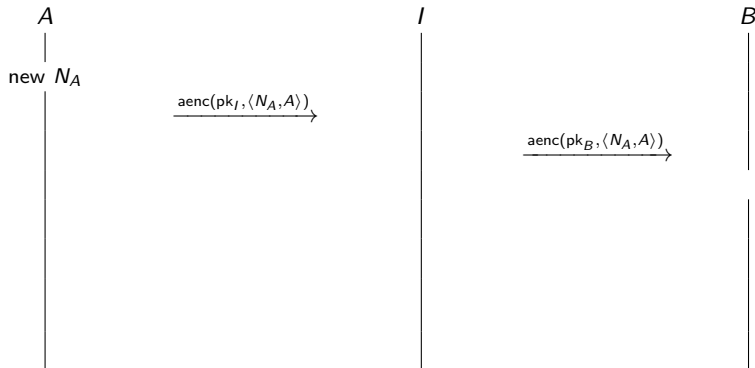
Attack found 17 years after the publication of the NS protocol!!



[G. Lowe. "An attack on the Needham-Schroeder public key authentication protocol". Information Processing Letters (November 1995)]

NSPK: Lowe's attack on authentication

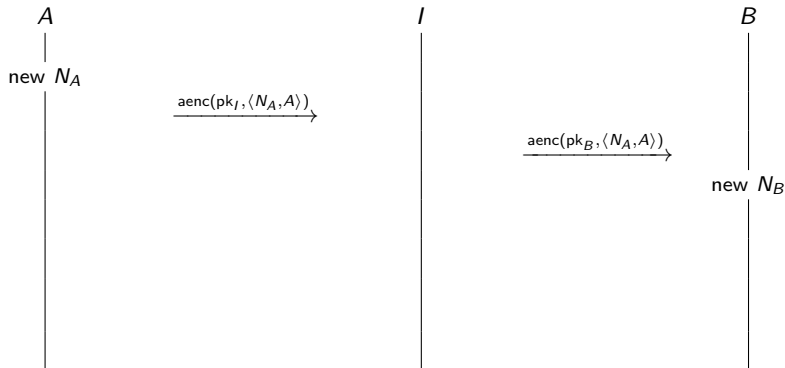
Attack found 17 years after the publication of the NS protocol!!



[G. Lowe. "An attack on the Needham-Schroeder public key authentication protocol". Information Processing Letters (November 1995)]

NSPK: Lowe's attack on authentication

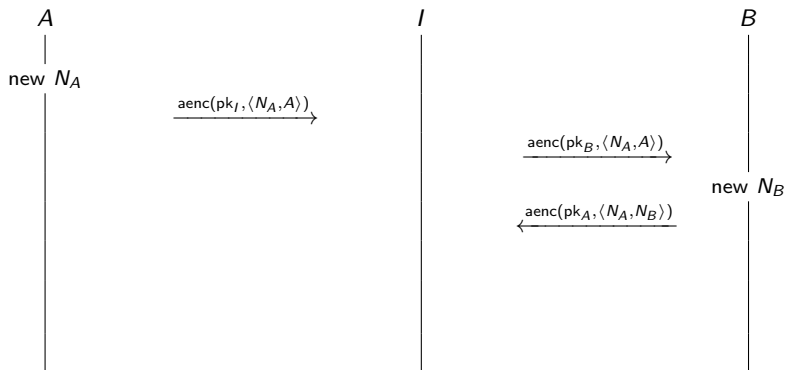
Attack found 17 years after the publication of the NS protocol!!



[G. Lowe. "An attack on the Needham-Schroeder public key authentication protocol". Information Processing Letters (November 1995)]

NSPK: Lowe's attack on authentication

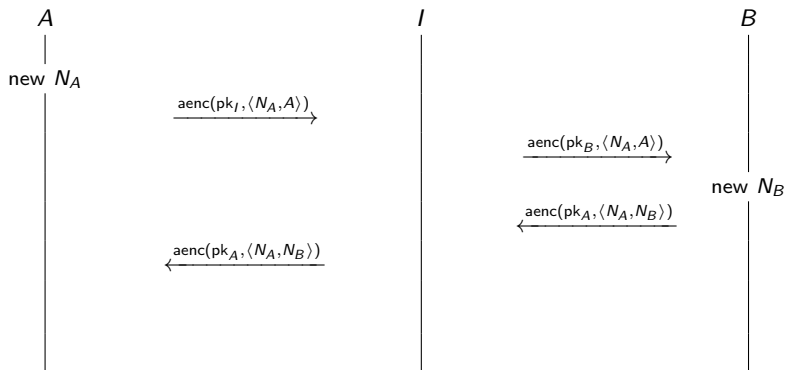
Attack found 17 years after the publication of the NS protocol!!



[G. Lowe. "An attack on the Needham-Schroeder public key authentication protocol". Information Processing Letters (November 1995)]

NSPK: Lowe's attack on authentication

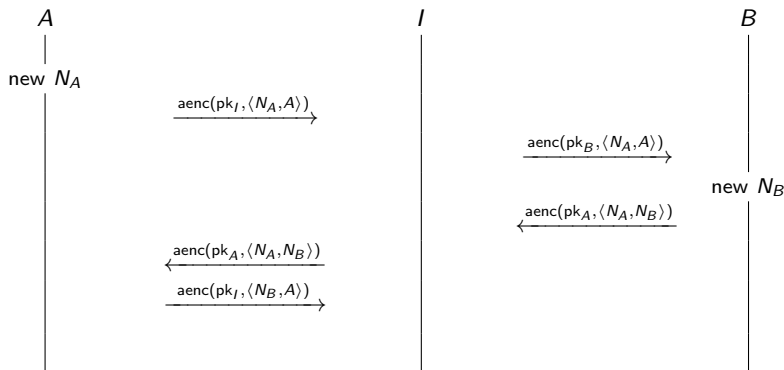
Attack found 17 years after the publication of the NS protocol!!



[G. Lowe. "An attack on the Needham-Schroeder public key authentication protocol". Information Processing Letters (November 1995)]

NSPK: Lowe's attack on authentication

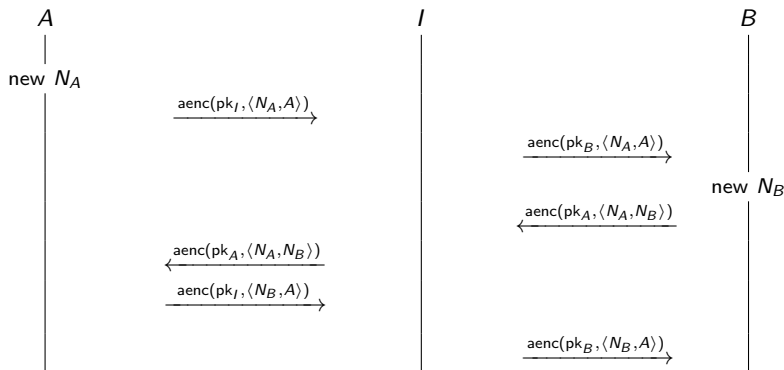
Attack found 17 years after the publication of the NS protocol!!



[G. Lowe. "An attack on the Needham-Schroeder public key authentication protocol". Information Processing Letters (November 1995)]

NSPK: Lowe's attack on authentication

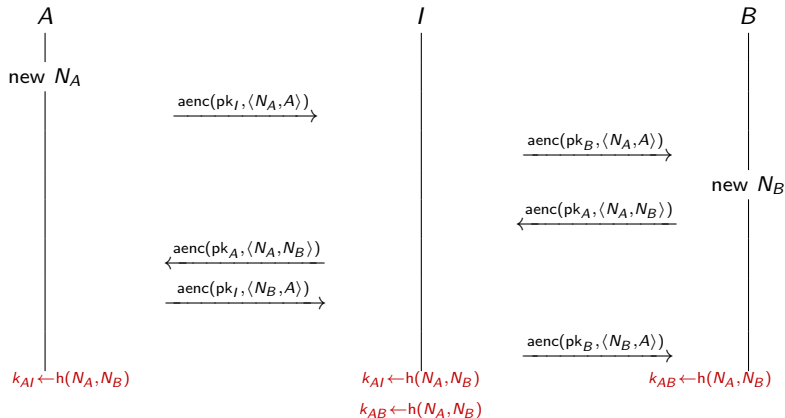
Attack found 17 years after the publication of the NS protocol!!



[G. Lowe. "An attack on the Needham-Schroeder public key authentication protocol". Information Processing Letters (November 1995)]

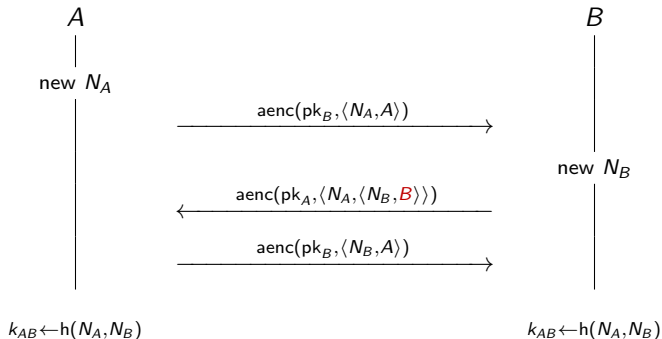
NSPK: Lowe's attack on authentication

Attack found 17 years after the publication of the NS protocol!!



[G. Lowe. "An attack on the Needham-Schroeder public key authentication protocol". Information Processing Letters (November 1995)]

NSPK: Lowe's fix



Forward secrecy

- ▶ The NSL protocol is secure against an attacker that controls the network.
- ▶ What if Alice's and Bob's private keys get compromised?
- ▶ What if the government forces Alice and Bob to reveal their private keys?
- ▶ Can we still protect confidentiality?

Forward secrecy

A protocol ensures **forward secrecy**, if even if long-term keys are compromised, past sessions of the protocol are still kept confidential, and this even if an attacker actively interfered.

The Station-to-Station (StS) protocol

A

|

|

B

|

|

The Station-to-Station (StS) protocol



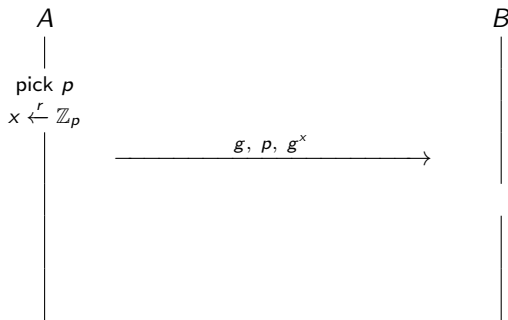
- ▶ where p is a large prime
- ▶ and g a generator of \mathbb{Z}_p^*

The Station-to-Station (StS) protocol



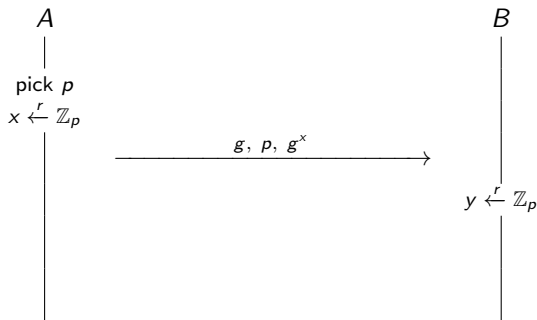
- ▶ where p is a large prime
- ▶ and g a generator of \mathbb{Z}_p^*

The Station-to-Station (StS) protocol



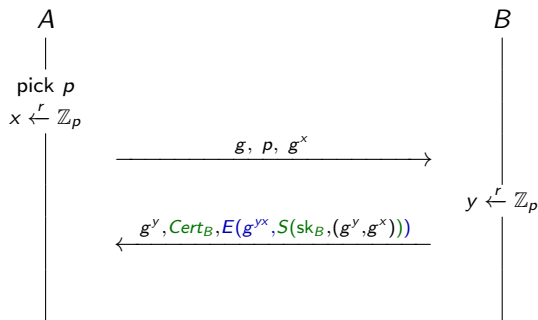
- ▶ where p is a large prime
- ▶ and g a generator of \mathbb{Z}_p^*

The Station-to-Station (StS) protocol



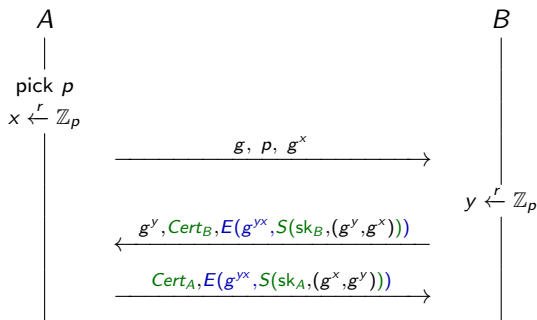
- ▶ where p is a large prime
- ▶ and g a generator of \mathbb{Z}_p^*

The Station-to-Station (StS) protocol



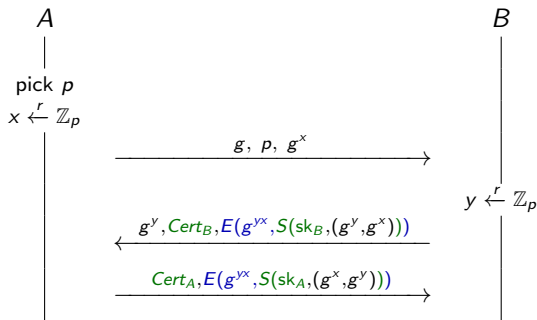
- ▶ where p is a large prime
- ▶ and g a generator of \mathbb{Z}_p^*

The Station-to-Station (StS) protocol



- ▶ where p is a large prime
- ▶ and g a generator of \mathbb{Z}_p^*

The Station-to-Station (StS) protocol



- ▶ where p is a large prime
- ▶ and g a generator of \mathbb{Z}_p^*

The StS ensures mutual authentication, key agreement, and forward secrecy