Classical Protocols
Needham-Schroeder Protocol Family

Design and Verification of Security Protocols and Security Ceremonies

Programa de Pós-Graduacão em Ciências da Computacão
Dr. Jean Everson Martina

March-June 2019

Programa de Pós-Graduação em Ciência da Computação
UFSC

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

Needham-Schroeder Shared Key Protocol

- Many existing protocols are derived from the work of Needham and Schroeder (1978);

Needham-Schroeder Shared Key Protocol

- Many existing protocols are derived from the work of Needham and Schroeder (1978);
- Kerberos authentication protocol suite is one of the main used protocols and is derived from NSSKP;

Needham-Schroeder Shared Key Protocol

- Many existing protocols are derived from the work of Needham and Schroeder (1978);
- Kerberos authentication protocol suite is one of the main used protocols and is derived from NSSKP;
- NSSKP is a shared-key authentication protocol designed to generate and propagate a session key which is used for subsequent symmetrically encrypted communication;

Needham-Schroeder Shared Key Protocol

- Many existing protocols are derived from the work of Needham and Schroeder (1978);
- Kerberos authentication protocol suite is one of the main used protocols and is derived from NSSKP;
- NSSKP is a shared-key authentication protocol designed to generate and propagate a session key which is used for subsequent symmetrically encrypted communication;
- There is no public key infrastructure in place.

NSSKP Goals

- The goal of the protocol is to establish mutual authentication between two parties A and B in the presence of adversary ;

NSSKP Goals

- The goal of the protocol is to establish mutual authentication between two parties A and B in the presence of adversary ;
- A and B obtain a secret shared key though authentication server S;

NSSKP Goals

- The goal of the protocol is to establish mutual authentication between two parties A and B in the presence of adversary ;
- A and B obtain a secret shared key though authentication server S;
- The adversary can intercept messages, delay messages, read and copy messages and generate messages;

NSSKP Goals

- The goal of the protocol is to establish mutual authentication between two parties A and B in the presence of adversary ;
- A and B obtain a secret shared key though authentication server S;
- The adversary can intercept messages, delay messages, read and copy messages and generate messages;
- The adversary can not learn the secret keys of principals, which they share with the authentication server S.

Assumptions of NSSKP

- There are three principals, two named A and B, desiring mutual authentication, and S, a trusted key server;

Assumptions of NSSKP

- There are three principals, two named A and B, desiring mutual authentication, and S, a trusted key server;
- It is assumed that A and B already have secure symmetric communication with S using keys K_{AS} and K_{BS} , respectively;

Assumptions of NSSKP

- There are three principals, two named A and B, desiring mutual authentication, and S, a trusted key server;
- It is assumed that A and B already have secure symmetric communication with S using keys K_{AS} and K_{BS} , respectively;
- It is assumed that the attacker can not be a legitimate party within the protocol.

Freshness Concepts of NSSKP

- NSSKP uses nonces which are randomly generated values included in messages and used only once;

Freshness Concepts of NSSKP

- NSSKP uses nonces which are randomly generated values included in messages and used only once;
- If a nonce is generated and sent by one agent in one step and returned by another in a later step, the generator knows that the message is fresh and not a replay from an earlier exchange;

Freshness Concepts of NSSKP

- NSSKP uses nonces which are randomly generated values included in messages and used only once;
- If a nonce is generated and sent by one agent in one step and returned by another in a later step, the generator knows that the message is fresh and not a replay from an earlier exchange;
- Note that a nonce is not anchored in time. The only assumption is that it has not been used in any earlier interchange, with high probability because it is random and not used twice.

How NSSKP works

- A makes contact with the authentication server S, sending identities A and B and nonce NA;

How NSSKP works

- A makes contact with the authentication server S, sending identities A and B and nonce NA;
- S responds with a message encrypted with the key of A. The message contains session key KAB (to be used by A and B) and certificate encrypted with B's key conveying the session key and A's identity;

How NSSKP works

- A makes contact with the authentication server S, sending identities A and B and nonce NA;
- S responds with a message encrypted with the key of A. The message contains session key KAB (to be used by A and B) and certificate encrypted with B's key conveying the session key and A's identity;
- A sends the certificate to B;

How NSSKP works

- A makes contact with the authentication server S, sending identities A and B and nonce NA;
- S responds with a message encrypted with the key of A. The message contains session key KAB (to be used by A and B) and certificate encrypted with B's key conveying the session key and A's identity;
- A sends the certificate to B;
- B decrypts the certificates and sends his own nonce encrypted by the session key to A; (nonce handshake);

How NSSKP works

- A makes contact with the authentication server S, sending identities A and B and nonce NA;
- S responds with a message encrypted with the key of A. The message contains session key KAB (to be used by A and B) and certificate encrypted with B's key conveying the session key and A's identity;
- A sends the certificate to B;
- B decrypts the certificates and sends his own nonce encrypted by the session key to A; (nonce handshake);
- A decrypts the last message and sends modified nonce back to B.

How NSSKP works

Goal

By the end of the message exchange both A and B share the secret key and both are assured in the presence of each other.

How to Understand Security Protocols

- There are questions to ask for any step in any protocol or ceremony:

How to Understand Security Protocols

- There are questions to ask for any step in any protocol or ceremony:
    - What is the sender trying to say with this message?

How to Understand Security Protocols

- There are questions to ask for any step in any protocol or ceremony:
  - What is the sender trying to say with this message?
  - What is the receiver entitled to believe after receiving the message?

How to Understand Security Protocols

- There are questions to ask for any step in any protocol or ceremony:
  - What is the sender trying to say with this message?
  - What is the receiver entitled to believe after receiving the message?
  - Can I use less resources to achieve the same goals?

How to Understand Security Protocols

- There are questions to ask for any step in any protocol or ceremony:
  - What is the sender trying to say with this message?
  - What is the receiver entitled to believe after receiving the message?
  - Can I use less resources to achieve the same goals?
  - Isn't there anything that I did not catch?

Notation For Protocol Description

A, B and S Agent names (Alice, Bob and Steve)

Notation For Protocol Description

A, B and S	Agent names (Alice, Bob and Steve)
N_A	Random number chosen by Alice (Nonce)

Notation For Protocol Description

A, B and S	Agent names (Alice, Bob and Steve)
N_A	Random number chosen by Alice (Nonce)
K_{AS}	Long term key shared between Alice and Steve

Notation For Protocol Description

A, B and S	Agent names (Alice, Bob and Steve)
N_A	Random number chosen by Alice (Nonce)
K_{AS}	Long term key shared between Alice and Steve
$\{X\}_{K_{AS}}$	Encrypted message using K_{AS}

NSSKP Message Exchange

1. A $\rightarrow$ S: A, B, N_A

NSSKP Message Exchange

1. A $\rightarrow$ S: A, B, N_A
2. S $\rightarrow$ A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

NSSKP Message Exchange

1. A → S: A, B, N_A
2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
3. A → B: $\{K_{AB}, A\}_{K_{BS}}$

NSSKP Message Exchange

1. A → S: A, B, N_A
2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
3. A → B: $\{K_{AB}, A\}_{K_{BS}}$
4. B → A: $\{N_B\}_{K_{AB}}$

NSSKP Message Exchange

1. A $\rightarrow$ S: A, B, N_A
2. S $\rightarrow$ A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
3. A $\rightarrow$ B: $\{K_{AB}, A\}_{K_{BS}}$
4. B $\rightarrow$ A: $\{N_B\}_{K_{AB}}$
5. A $\rightarrow$ B: $\{N_B - 1\}_{K_{AB}}$

NSSKP - What is being said!

1. A → S: A, B, N_A

NSSKP - What is being said!

1. A → S: A, B, N_A
   Hi Steve, this is Alice, I want to talk to Bob, that's the identifier of my request.

NSSKP - What is being said!

1. A → S: A, B, N_A
   Hi Steve, this is Alice, I want to talk to Bob, that's the identifier of my request.
2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

NSSKP - What is being said!

1. A $\rightarrow$ S: A, B, N_A
   Hi Steve, this is Alice, I want to talk to Bob, that's the identifier of my request.
2. S $\rightarrow$ A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
   Alice I am sending you a secret which shows your identifier, Bob's identity and the key for you to talk to him. Here is a ticket to send Bob the key and relate it to your identity.

NSSKP - What is being said!

1. A → S: A, B, N_A
   Hi Steve, this is Alice, I want to talk to Bob, that's the identifier of my request.
2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
   Alice I am sending you a secret which shows your identifier, Bob's identity and the key for you to talk to him. Here is a ticket to send Bob the key and relate it to your identity.
3. A → B: $\{K_{AB}, A\}_{K_{BS}}$

NSSKP - What is being said!

1. A → S: A, B, N_A
   Hi Steve, this is Alice, I want to talk to Bob, that's the identifier of my request.
2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
   Alice I am sending you a secret which shows your identifier, Bob's identity and the key for you to talk to him. Here is a ticket to send Bob the key and relate it to your identity.
3. A → B: $\{K_{AB}, A\}_{K_{BS}}$
   Bob there is a ticket for you!

NSSKP - What is being said!

4. B → A: $\{N_B\}_{K_{AB}}$

NSSKP - What is being said!

4. B → A: $\{N_B\}_{K_{AB}}$
   I want to challenge Alice to see if she has the key on the ticket.

NSSKP - What is being said!

4. B → A: $\{N_B\}_{K_{AB}}$
   I want to challenge Alice to see if she has the key on the ticket.
5. A → B: $\{N_B - 1\}_{K_{AB}}$

NSSKP - What is being said!

4. B → A: $\{N_B\}_{K_{AB}}$
   I want to challenge Alice to see if she has the key on the ticket.
5. A → B: $\{N_B - 1\}_{K_{AB}}$
   Challenge accepted. Take it back!

NSSKP Knowledge Exchange

Alice Knows N_A and K_{AS}

NSSKP Knowledge Exchange

Alice Knows N_A and K_{AS}

1. A → S: A, B, N_A

NSSKP Knowledge Exchange

Alice Knows N_A and K_{AS}

1. A → S: A, B, N_A

Steve Knows N_A, K_{AB}, K_{BS} and K_{AS}

NSSKP Knowledge Exchange

Alice Knows N_A and K_{AS}

1. A → S: A, B, N_A

   Steve Knows N_A, K_{AB}, K_{BS} and K_{AS}

2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

NSSKP Knowledge Exchange

Alice Knows N_A and K_{AS}

1. A → S: A, B, N_A

   Steve Knows N_A, K_{AB}, K_{BS} and K_{AS}

2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

   Alice Knows N_A, K_{AS} and K_{AB}

NSSKP Knowledge Exchange

Alice Knows N_A and K_{AS}

1. A → S: A, B, N_A

   Steve Knows N_A, K_{AB}, K_{BS} and K_{AS}

2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

   Alice Knows N_A, K_{AS} and K_{AB}

3. A → B: $\{K_{AB}, A\}_{K_{BS}}$

NSSKP Knowledge Exchange

Alice Knows N_A and K_{AS}

1. A → S: A, B, N_A

   Steve Knows N_A, K_{AB}, K_{BS} and K_{AS}

2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

   Alice Knows N_A, K_{AS} and K_{AB}

3. A → B: $\{K_{AB}, A\}_{K_{BS}}$

   Bob Knows N_B, K_{BS} and K_{AB}

NSSKP Knowledge Exchange

Alice Knows N_A and K_{AS}

1. A → S: A, B, N_A

   Steve Knows N_A, K_{AB}, K_{BS} and K_{AS}

2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

   Alice Knows N_A, K_{AS} and K_{AB}

3. A → B: $\{K_{AB}, A\}_{K_{BS}}$

   Bob Knows N_B, K_{BS} and K_{AB}

4. B → A: $\{N_B\}_{K_{AB}}$

NSSKP Knowledge Exchange

Alice Knows N_A and K_{AS}

1. A → S: A, B, N_A

   Steve Knows N_A, K_{AB}, K_{BS} and K_{AS}

2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

   Alice Knows N_A, K_{AS} and K_{AB}

3. A → B: $\{K_{AB}, A\}_{K_{BS}}$

   Bob Knows N_B, K_{BS} and K_{AB}

4. B → A: $\{N_B\}_{K_{AB}}$

   Alice Knows N_A, N_B, K_{AS} and K_{AB}

NSSKP Knowledge Exchange

Alice Knows N_A and K_{AS}

1. A → S: A, B, N_A

Steve Knows N_A, K_{AB}, K_{BS} and K_{AS}

2. S → A: $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

Alice Knows N_A, K_{AS} and K_{AB}

3. A → B: $\{K_{AB}, A\}_{K_{BS}}$

Bob Knows N_B, K_{BS} and K_{AB}

4. B → A: $\{N_B\}_{K_{AB}}$

Alice Knows N_A, N_B, K_{AS} and K_{AB}

5. A → B: $\{N_B - 1\}_{K_{AB}}$

Informal Verification of Security Protocols

- What to ask about a protocol or ceremony:

Informal Verification of Security Protocols

- What to ask about a protocol or ceremony:
    - Are both authentication and secrecy assured?

Informal Verification of Security Protocols

- What to ask about a protocol or ceremony:
    - Are both authentication and secrecy assured?
    - Is it possible to impersonate one or more of the parties?

Informal Verification of Security Protocols

- What to ask about a protocol or ceremony:
  - Are both authentication and secrecy assured?
  - Is it possible to impersonate one or more of the parties?
  - Is it possible to interject messages from an earlier exchange (replay attack)?

Informal Verification of Security Protocols

- What to ask about a protocol or ceremony:
    - Are both authentication and secrecy assured?
    - Is it possible to impersonate one or more of the parties?
    - Is it possible to interject messages from an earlier exchange (replay attack)?
    - What tools can an attacker deploy?

Informal Verification of Security Protocols

- What to ask about a protocol or ceremony:
  - Are both authentication and secrecy assured?
  - Is it possible to impersonate one or more of the parties?
  - Is it possible to interject messages from an earlier exchange (replay attack)?
  - What tools can an attacker deploy?
  - If any key is compromised, what are the consequences?

Denning and Sacco Attack

- Denning and Sacco pointed out that the compromise of a session key has bad consequences. An intruder can reuse an old session key and pass it off as a new one;

Denning and Sacco Attack

- Denning and Sacco pointed out that the compromise of a session key has bad consequences. An intruder can reuse an old session key and pass it off as a new one;
- Suppose Charlie has acquired K_{AB} from a past run of the protocol, and start the protocol from message 3;

Denning and Sacco Attack

- Denning and Sacco pointed out that the compromise of a session key has bad consequences. An intruder can reuse an old session key and pass it off as a new one;
- Suppose Charlie has acquired K_{AB} from a past run of the protocol, and start the protocol from message 3;

Bob will believe he is talking to Alice

Denning and Sacco Attack

- Denning and Sacco pointed out that the compromise of a session key has bad consequences. An intruder can reuse an old session key and pass it off as a new one;
- Suppose Charlie has acquired K_{AB} from a past run of the protocol, and start the protocol from message 3;

Bob will believe he is talking to Alice

3. C $\rightarrow$ B: $\{K_{AB}, A\}_{K_{BS}}$

Denning and Sacco Attack

- Denning and Sacco pointed out that the compromise of a session key has bad consequences. An intruder can reuse an old session key and pass it off as a new one;
- Suppose Charlie has acquired K_{AB} from a past run of the protocol, and start the protocol from message 3;

Bob will believe he is talking to Alice

3. C $\rightarrow$ B: $\{K_{AB}, A\}_{K_{BS}}$
4. B $\rightarrow$ C: $\{N_B\}_{K_{AB}}$

Denning and Sacco Attack

- Denning and Sacco pointed out that the compromise of a session key has bad consequences. An intruder can reuse an old session key and pass it off as a new one;
- Suppose Charlie has acquired K_{AB} from a past run of the protocol, and start the protocol from message 3;

Bob will believe he is talking to Alice

3. C $\rightarrow$ B: $\{K_{AB}, A\}_{K_{BS}}$
4. B $\rightarrow$ C: $\{N_B\}_{K_{AB}}$
5. C $\rightarrow$ B: $\{N_B - 1\}_{K_{AB}}$

Denning and Sacco Attack

- Message 3 is not protected by any freshness component;

Denning and Sacco Attack

- Message 3 is not protected by any freshness component;
- There is no way for B to know if the K_{AB} it receives is current;

Denning and Sacco Attack

- Message 3 is not protected by any freshness component;
- There is no way for B to know if the K_{AB} it receives is current;
- Lack of freshness on message 3 means an intruder has unlimited time to crack an old session key and reuse it.

Bauer, et al. Attack on NSSKP

- Bauer, et al. pointed out that if key K_{AS} were compromised, anyone could impersonate A and establish communication with any other party;

Bauer, et al. Attack on NSSKP

- Bauer, et al. pointed out that if key K_{AS} were compromised, anyone could impersonate A and establish communication with any other party;
- Usually the lost of control on long term secrets affects deeply how a protocol operate;

Bauer, et al. Attack on NSSKP

- Bauer, et al. pointed out that if key K_{AS} were compromised, anyone could impersonate A and establish communication with any other party;
- Usually the lost of control on long term secrets affects deeply how a protocol operate;
- It is important to have mechanisms that could revoke keys or at least render them unusable after sometime.

Questions!

- These flaws persisted for almost 10 years before they were discovered. Why did it take that long to see that?

Questions!

- These flaws persisted for almost 10 years before they were discovered. Why did it take that long to see that?
- Ask what happens if a key is broken is a fair question?

Questions!

- These flaws persisted for almost 10 years before they were discovered. Why did it take that long to see that?
- Ask what happens if a key is broken is a fair question?
- How can you address these design faults pointed out by Denning and Sacco and Bauer et al.?

Needham-Schroeder Public Key Protocol

- Many existing protocols are derived from the work of Needham and Schroeder (1978);

Needham-Schroeder Public Key Protocol

- Many existing protocols are derived from the work of Needham and Schroeder (1978);
- The handshake done by many public key cryptographic protocols are derived from NSPKP;

Needham-Schroeder Public Key Protocol

- Many existing protocols are derived from the work of Needham and Schroeder (1978);
- The handshake done by many public key cryptographic protocols are derived from NSPKP;
- NSPKP is a public-key authentication protocol designed to generate and propagate a session key which is used for subsequent symmetrically encrypted communication;

Needham-Schroeder Public Key Protocol

- Many existing protocols are derived from the work of Needham and Schroeder (1978);
- The handshake done by many public key cryptographic protocols are derived from NSPKP;
- NSPKP is a public-key authentication protocol designed to generate and propagate a session key which is used for subsequent symmetrically encrypted communication;
- There is no public key infrastructure in place, but the identities related top public keys are an assumption.

Needham-Schroeder Public Key Protocol

1. A → B: $\{|N_a, A|\}_{K_b}$

Needham-Schroeder Public Key Protocol

1. A $\rightarrow$ B: $\{|N_a, A|\}_{K_b}$
2. B $\rightarrow$ A: $\{|N_a, N_b|\}_{K_a}$

Needham-Schroeder Public Key Protocol

1. A → B: $\{|N_a, A|\}_{K_b}$
2. B → A: $\{|N_a, N_b|\}_{K_a}$
3. A → B: $\{|N_b|\}_{K_b}$

NSPKP Goals

- The goal of the protocol is to establish mutual authentication between two parties A and B in the presence of adversary;

NSPKP Goals

- The goal of the protocol is to establish mutual authentication between two parties A and B in the presence of adversary;
- A and B obtain a secret shared key though direct communication using public key cryptography;

NSPKP Goals

- The goal of the protocol is to establish mutual authentication between two parties A and B in the presence of adversary;
- A and B obtain a secret shared key though direct communication using public key cryptography;
- This adversary can intercept messages, delay messages, read and copy messages and generate messages;

NSPKP Goals

- The goal of the protocol is to establish mutual authentication between two parties A and B in the presence of adversary;
- A and B obtain a secret shared key though direct communication using public key cryptography;
- This adversary can intercept messages, delay messages, read and copy messages and generate messages;
- This adversary can not learn the privates keys of principals.

Freshness Concepts of NSPKP

- NSPKP uses nonces which are randomly generated values included in messages and used only once;

Freshness Concepts of NSPKP

- NSPKP uses nonces which are randomly generated values included in messages and used only once;
- If a nonce is generated and sent by one agent in one step and returned by another in a later step, the generator knows that the message is fresh and not a replay from an earlier exchange;

Freshness Concepts of NSPKP

- NSPKP uses nonces which are randomly generated values included in messages and used only once;
- If a nonce is generated and sent by one agent in one step and returned by another in a later step, the generator knows that the message is fresh and not a replay from an earlier exchange;
- Note that a nonce is not anchored in time. The only assumption is that it has not been used in any earlier interchange, with high probability because it is random and not used twice.

Needham-Schroeder Public Key Protocol - Questions

- What are the assumptions?

Needham-Schroeder Public Key Protocol - Questions

- What are the assumptions?
- What seems to be the goal?

Needham-Schroeder Public Key Protocol - Questions

- What are the assumptions?
- What seems to be the goal?
- What might the principals believe after each step?

Needham-Schroeder Public Key Protocol - Questions

- What are the assumptions?
- What seems to be the goal?
- What might the principals believe after each step?
- What is missing?

Needham-Schroeder Public Key Protocol - Questions

- What are the assumptions?
- What seems to be the goal?
- What might the principals believe after each step?
- What is missing?
- Is it secure?

Needham-Schroeder Public Key Protocol Description

1. A → B: $\{|N_a, A|\}_{K_b}$
   Alice send to Bob an encrypted nonce

Needham-Schroeder Public Key Protocol Description

1. A $\rightarrow$ B: $\{|N_a, A|\}_{K_b}$
   Alice send to Bob an encrypted nonce
2. B $\rightarrow$ A: $\{|N_a, N_b|\}_{K_a}$
   Bob returns to Alice her Nonce together with his own

Needham-Schroeder Public Key Protocol Description

1. A → B: $\{|N_a, A|\}_{K_b}$
   Alice send to Bob an encrypted nonce
2. B → A: $\{|N_a, N_b|\}_{K_a}$
   Bob returns to Alice her Nonce together with his own
3. A → B: $\{|N_b|\}_{K_b}$
   Alice returns to Bob his Nonce

Needham-Schroeder Public Key Protocol Assumptions

1. A → B: $\{|N_a, A|\}_{K_b}$
   Only Alice knows Na before message 1
   Only Bob can decrypt message 1

Needham-Schroeder Public Key Protocol Assumptions

1. A → B: $\{|N_a, A|\}_{K_b}$
   Only Alice knows Na before message 1
   Only Bob can decrypt message 1
2. B → A: $\{|N_a, N_b|\}_{K_a}$
   Only Bob knows Nb before message 2
   Bob knows Na because he can decrypt
   Ohly Alice can decrypt message 2

Needham-Schroeder Public Key Protocol Assumptions

1. A → B: $\{|N_a, A|\}_{K_b}$
   Only Alice knows Na before message 1
   Only Bob can decrypt message 1
2. B → A: $\{|N_a, N_b|\}_{K_a}$
   Only Bob knows Nb before message 2
   Bob knows Na because he can decrypt
   Ohly Alice can decrypt message 2
3. A → B: $\{|N_b|\}_{K_b}$
   Alice knows Nb because she can decrypt
   Only Bob can decrypt message 3

Needham-Schroeder Public Key Protocol Assumptions

1. A → B: $\{|N_a, A|\}_{K_b}$
   Only Alice knows Na before message 1
   Only Bob can decrypt message 1
2. B → A: $\{|N_a, N_b|\}_{K_a}$
   Only Bob knows Nb before message 2
   Bob knows Na because he can decrypt
   Ohly Alice can decrypt message 2
3. A → B: $\{|N_b|\}_{K_b}$
   Alice knows Nb because she can decrypt
   Only Bob can decrypt message 3
   Why do we need message 3?

Needham-Schroeder Public Key Protocol Interpretation

1. A $\rightarrow$ B: $\{|N_a, A|\}_{K_b}$
   Alice starts a session Na is the session control
   Alice's identity (A) is for Bob to know whom
   to encrypt message 2

Needham-Schroeder Public Key Protocol Interpretation

1. A → B: $\{|N_a, A|\}_{K_b}$
   Alice starts a session Na is the session control
   Alice's identity (A) is for Bob to know whom
   to encrypt message 2
2. B → A: $\{|N_a, N_b|\}_{K_a}$
   Bom sens Na back to keep the session
   Bob uses Nb to authenticate Alice on message 3
   When receiving message 2 Alice knows that only Bob could
   have created it because it contains Na

Needham-Schroeder Public Key Protocol Interpretation

1. A → B: $\{|N_a, A|\}_{K_b}$
   Alice starts a session Na is the session control
   Alice's identity (A) is for Bob to know whom
   to encrypt message 2
2. B → A: $\{|N_a, N_b|\}_{K_a}$
   Bom sens Na back to keep the session
   Bob uses Nb to authenticate Alice on message 3
   When receiving message 2 Alice knows that only Bob could
   have created it because it contains Na
3. A → B: $\{|N_b|\}_{K_b}$
   Alice already authenticated Bob. Now she wants to authenticat
   When receiving message 3 Bob knows that only Alice could
   have created it because it contains Nb

Needham-Schroeder Public Key Protocol Objectives

1. A → B: $\{|N_a, A|\}_{K_b}$
2. B → A: $\{|N_a, N_b|\}_{K_a}$
3. A → B: $\{|N_b|\}_{K_b}$

- The protocols authenticates Alice to Bob;

Needham-Schroeder Public Key Protocol Objectives

1. A → B: $\{|N_a, A|\}_{K_b}$
2. B → A: $\{|N_a, N_b|\}_{K_a}$
3. A → B: $\{|N_b|\}_{K_b}$

- The protocols authenticates Alice to Bob;
- The protocols authenticate Bob to Alice;

Needham-Schroeder Public Key Protocol Objectives

1. A → B: $\{|N_a, A|\}_{K_b}$
2. B → A: $\{|N_a, N_b|\}_{K_a}$
3. A → B: $\{|N_b|\}_{K_b}$

- The protocols authenticates Alice to Bob;
- The protocols authenticate Bob to Alice;
- By the usage of fresh Nonces, we obtain the aliveness property in the protocols;

Needham-Schroeder Public Key Protocol Objectives

1. A → B: $\{|N_a, A|\}_{K_b}$
2. B → A: $\{|N_a, N_b|\}_{K_a}$
3. A → B: $\{|N_b|\}_{K_b}$

- The protocols authenticates Alice to Bob;
- The protocols authenticate Bob to Alice;
- By the usage of fresh Nonces, we obtain the aliveness property in the protocols;
- But, Is it secure?

Almighty Attack - Rules of the Game

- Charlie is a very powerful attacker.

Almighty Attack - Rules of the Game

- Charlie is a very powerful attacker.
- He can:

Almighty Attack - Rules of the Game

- Charlie is a very powerful attacker.
- He can:
    - Intercept anything in the network;

Almighty Attack - Rules of the Game

- Charlie is a very powerful attacker.
- He can:
    - Intercept anything in the network;
    - Block anything in the network;

Almighty Attack - Rules of the Game

- Charlie is a very powerful attacker.
- He can:
    - Intercept anything in the network;
    - Block anything in the network;
    - Replay Messages;

Almighty Attack - Rules of the Game

- Charlie is a very powerful attacker.
- He can:
  - Intercept anything in the network;
  - Block anything in the network;
  - Replay Messages;
  - Create new messages with what he learned with the traffic;

Almighty Attack - Rules of the Game

- Charlie is a very powerful attacker.
- He can:
  - Intercept anything in the network;
  - Block anything in the network;
  - Replay Messages;
  - Create new messages with what he learned with the traffic;
  - Behave as a normal agent;

Almighty Attack - Rules of the Game

- Charlie is a very powerful attacker.
- He can:
  - Intercept anything in the network;
  - Block anything in the network;
  - Replay Messages;
  - Create new messages with what he learned with the traffic;
  - Behave as a normal agent;
- He can not:

Almighty Attack - Rules of the Game

- Charlie is a very powerful attacker.
- He can:
    - Intercept anything in the network;
    - Block anything in the network;
    - Replay Messages;
    - Create new messages with what he learned with the traffic;
    - Behave as a normal agent;
- He can not:
    - Break cryptography;

Almighty Attack - Rules of the Game

- Charlie is a very powerful attacker.
- He can:
  - Intercept anything in the network;
  - Block anything in the network;
  - Replay Messages;
  - Create new messages with what he learned with the traffic;
  - Behave as a normal agent;
- He can not:
  - Break cryptography;
  - Guess random numbers.

Almighty Attack - Lowe's Attack

1. A $\rightarrow$ C: $\{|N_a, A|\}_{K_c}$

Almighty Attack - Lowe's Attack

1. A → C: $\{|N_a, A|\}_{K_c}$

1'. C(A) → B: $\{|N_a, A|\}_{K_b}$

2'. B → C(A): $\{|N_a, N_b|\}_{K_a}$

Almighty Attack - Lowe's Attack

1. A → C: $\{|N_a, A|\}_{K_c}$

1'. C(A) → B: $\{|N_a, A|\}_{K_b}$

2'. B → C(A): $\{|N_a, N_b|\}_{K_a}$

2. C → A: $\{|N_a, N_b|\}_{K_a}$

Almighty Attack - Lowe's Attack

1. A → C: $\{|N_a, A|\}_{K_c}$
1'. C(A) → B: $\{|N_a, A|\}_{K_b}$
2'. B → C(A): $\{|N_a, N_b|\}_{K_a}$
2. C → A: $\{|N_a, N_b|\}_{K_a}$
3. A → C: $\{|N_b|\}_{K_c}$

Almighty Attack - Lowe's Attack

1. A → C: $\{|N_a, A|\}_{K_c}$

1'. C(A) → B: $\{|N_a, A|\}_{K_b}$

2'. B → C(A): $\{|N_a, N_b|\}_{K_a}$

2. C → A: $\{|N_a, N_b|\}_{K_a}$

3. A → C: $\{|N_b|\}_{K_c}$

3'. C → B: $\{|N_b|\}_{K_b}$

- Bob believes to be talking to Alice , while he is talking to Charlie;

Almighty Attack - Lowe's Attack

1. A → C: $\{|N_a, A|\}_{K_c}$

1'. C(A) → B: $\{|N_a, A|\}_{K_b}$

2'. B → C(A): $\{|N_a, N_b|\}_{K_a}$

2. C → A: $\{|N_a, N_b|\}_{K_a}$

3. A → C: $\{|N_b|\}_{K_c}$

3'. C → B: $\{|N_b|\}_{K_b}$

- Bob believes to be talking to Alice , while he is talking to Charlie;
- Charlie uses Alice as an oracle to answers Bob's challenges;

Almighty Attack - Lowe's Attack

1. A → C: $\{|N_a, A|\}_{K_c}$
1'. C(A) → B: $\{|N_a, A|\}_{K_b}$
2'. B → C(A): $\{|N_a, N_b|\}_{K_a}$
2. C → A: $\{|N_a, N_b|\}_{K_a}$
3. A → C: $\{|N_b|\}_{K_c}$
3'. C → B: $\{|N_b|\}_{K_b}$

- Bob believes to be talking to Alice , while he is talking to Charlie;
- Charlie uses Alice as an oracle to answers Bob's challenges;
- Charlie can use Nb to prove to Bob he is Alice.

Lowe's Attack - Facts

- Gavin Lowe was a random Computer Theoretician at Oxford;

Lowe’s Attack - Facts

- Gavin Lowe was a random Computer Theoretician at Oxford;
- The attack looks easy, but it took 15 years to be found;

Lowe's Attack - Facts

- Gavin Lowe was a random Computer Theoretician at Oxford;
- The attack looks easy, but it took 15 years to be found;
- The attack works because of a change on the threat model;

Lowe's Attack - Facts

- Gavin Lowe was a random Computer Theoretician at Oxford;
- The attack looks easy, but it took 15 years to be found;
- The attack works because of a change on the threat model;
- But his attack is important because it was only discovered with the help of a formal verification tool.

How Lowe did it?

- He used a tool called a model checker;

How Lowe did it?

- He used a tool called a model checker;
- This tool has able to saturate the protocol execution;

How Lowe did it?

- He used a tool called a model checker;
- This tool has able to saturate the protocol execution;
- The idea is to explore all the reachable states of the model;

How Lowe did it?

- He used a tool called a model checker;
- This tool has able to saturate the protocol execution;
- The idea is to explore all the reachable states of the model;
- This verification is considered bound to the amount of peers and parallel runs tested.

What we learned from this example?

- Some tests are too hard to be executed by hand;

What we learned from this example?

- Some tests are too hard to be executed by hand;
- Theoretical tools are a good breakthrough for any area;

What we learned from this example?

- Some tests are too hard to be executed by hand;
- Theoretical tools are a good breakthrough for any area;
- Even very simple and well studied protocols may contain hidden failures;

What we learned from this example?

- Some tests are too hard to be executed by hand;
- Theoretical tools are a good breakthrough for any area;
- Even very simple and well studied protocols may contain hidden failures;
- We learned to be diligent and somewhat paranoid on protocols and how they achieve their goals.

Discussion

- Which other protocols do you believe there are failures that were not detected yet?

Discussion

- Which other protocols do you believe there are failures that were not detected yet?
- How do we correct Lowe's attack on NSPKP?

Discussion

- Which other protocols do you believe there are failures that were not detected yet?
- How do we correct Lowe's attack on NSPKP?
- Is formally proving a protocol enough to claim it is secure?

Discussion

- Which other protocols do you believe there are failures that were not detected yet?
- How do we correct Lowe's attack on NSPKP?
- Is formally proving a protocol enough to claim it is secure?
- What if a user drop an assumption of the protocol? Is it still secure?

Discussion

- Which other protocols do you believe there are failures that were not detected yet?
- How do we correct Lowe's attack on NSPKP?
- Is formally proving a protocol enough to claim it is secure?
- What if a user drop an assumption of the protocol? Is it still secure?
- How secure is formally secure?

Questions????

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

UNIVERSIDADE FEDERAL
DE SANTA CATARINA