A Logic of Authentication

### 2020310795 康鸿博

## What are the problems mentioned by the paper? (intro)

In this paper, the authors point out a problem: it’s hard to design an authentication protocol that survive from different attacks, especially when people want to achieve better performance by reducing the data sent or calculated as much as possible. The difficulties exist in different aspects: (1) The initial data (e.g. whether you know the public key of another node, or you need to get that from a trusted server) and the basic assumptions (e.g. whether there’s a synchronized clock) hold and trusted by principals are different because of different network setting. (2) The protocol designer must prohibit “any” possible attack to the protocol, especially those not found clearly stated yet. This means that it’s inadequate for the protocol designer to use a “black list” approach by saying the protocol is protected from some attacks, but they need to generate a “white list” proof for the protocol.

The authors give concrete analysis for four commonly used authentication protocols based on its predicate logic, and show how common variations lead to potential attacks.

## Summary of major innovations (intro)

This is the very first paper to analyze authentication protocols using simple logic. It shows good results by proving correctness in some widely used protocols, and shows flaws in some of their variations. These flaws are also identified and fixed by sporadic literature, but this is the first paper to give a theoretical and systematic analysis of authentication protocols about correctness and succinctness.

## How about the important related works/papers? (related work)

Four widely-used authentication protocols are analyzed in the paper as example: Kerberos Protocol, Andrew Secure RPC Handshake, Needham-Schroeder Public-Key Protocol, CCITT X.509 Protocol. These algorithms have different assumptions and have different design targets:

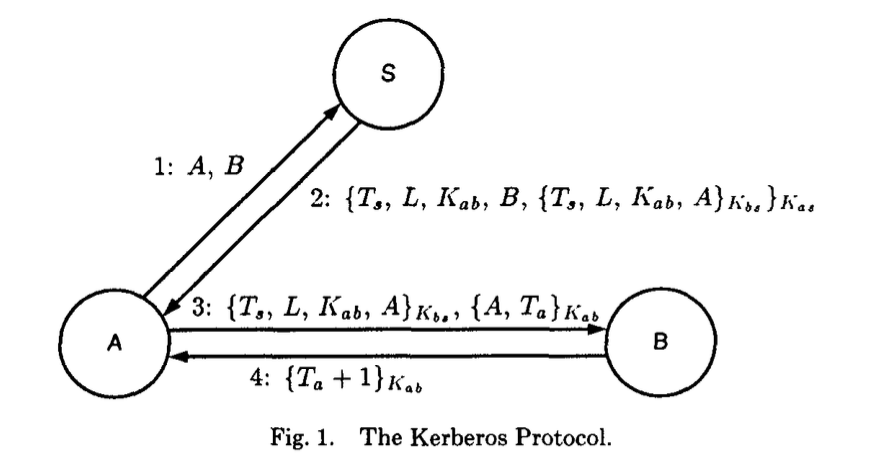
1. The design target of Kerberos Protocol is similar to that of the “RSA key exchange” protocol: to build an encrypted message channel between two servers. Kerberos use a trusted third party, and it prohibit the man-in-the-middle attack.
2. The design target of Andrew Secure RPC Handshake is to get a new session key when two servers already have a key.
3. The design target of Needham-Schroeder Public-Key Protocol is to allow two servers, each with their own public key and private key, to exchange two secret number, under the assistance of a trusted server.
4. The design target of CCITT X.509 Protocol is to send Xa, Ya from A to B, and send Xb, Yb from B to A, ensuring the integrity of Xa and Xb, assuring the recipient of their origin, and guarantees the privacy of Ya and Yb.

## What are some intriguing aspects of the paper? (design & implementation)

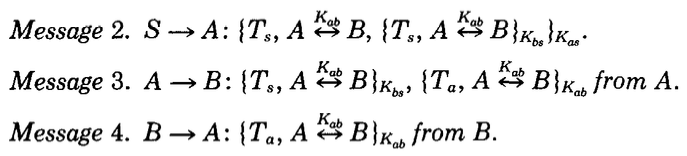
###### Predicates:

1. P believe X: P believe that X is true
2. P sees X: P receive message about X
3. P said X: P sent message about X
4. P controls X: P has an authority on X, this generally means that P is responsible for (the generation) of X.
5. Fresh(X): X is a fresh message from this execution of the protocol. This help prevent replay attack.
6. P <-K-> Q: P and Q use an exclusive key to communicate. Note that trusted principals may also have this key.
   1. Similar:  P and Q have a shared secret X. It may not be a key. Trusted principals may also have this key.
7. |-K-> P: K is P’s public key.
8. {X}K :X is encrypted by K. It’s full form is ({X}K from P) to prevent P from receiving this message.
   1. <X>Y : X is combined with Y, which is a secret

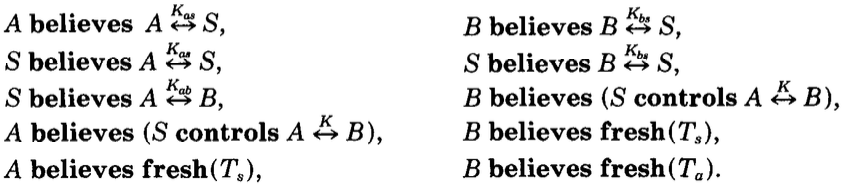
###### Kerberos Protocol:

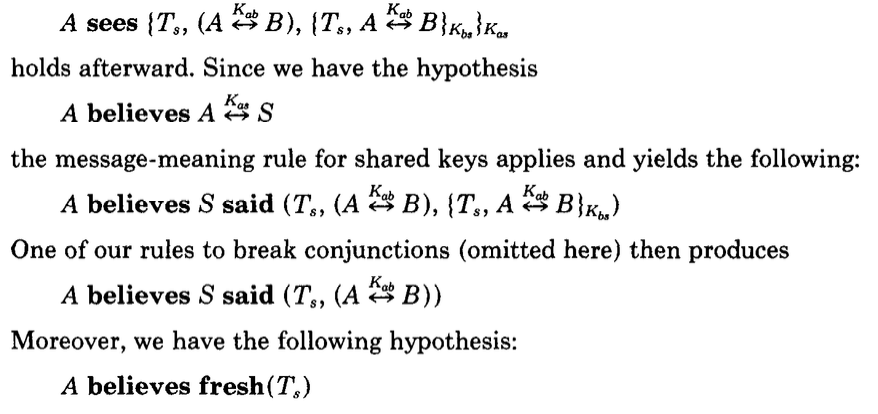


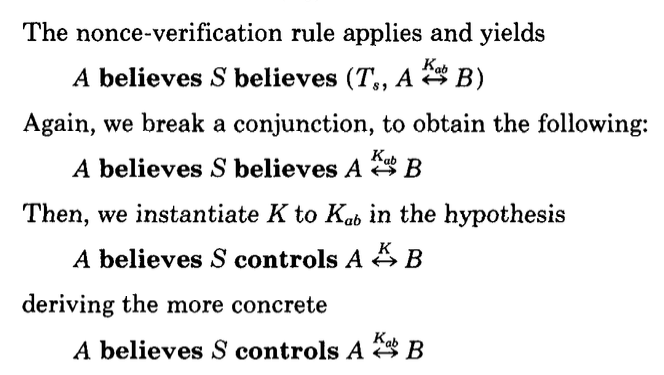
Idealized model:

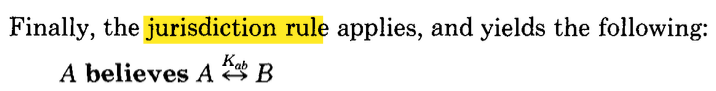


Direct Assumption:

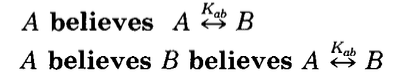








One interesting example is that, if we don’t send the first 3 messages instead of all 4 messages, we can still get



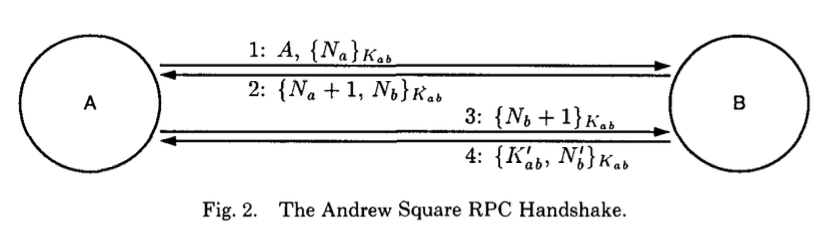


but not

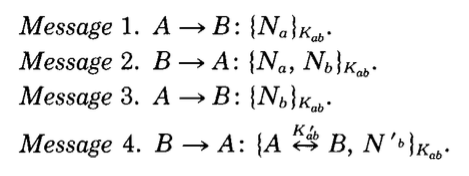


It not obvious what may happen in this case, but actually, what may happen here is that B may not exist. The reason that the 4th statement doesn’t hold is that A doesn’t believe in B’s existence.

###### Andrew Secure RPC Handshake:

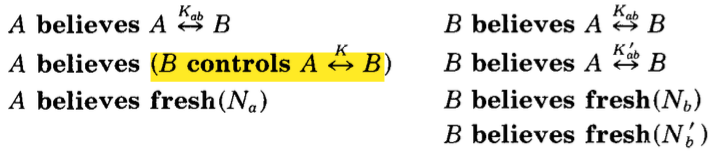


Idealized Model:

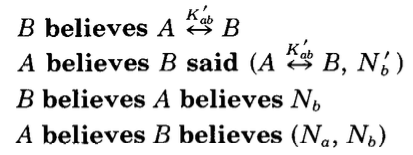


Note that in the 4th message, N’b is an unrelated value used in the future.

Assumption:



Deduced result:



Note that we can’t derive the mutual belief of A and B on the channel with key K’ab. This incorrectness leads to a reply attack on the 4th message. A man-in-the-middle can resend the 4th message to start a connection with A.

## How can the research be improved? (the bad side, future work, your idea)

The result isn’t obvious. When theoretical deduction tells that the mutual belief doesn’t hold, we can’t easily find an attack to the system.

After the development of automated verification, we may be able to verify network protocols by a model checker.

## Give the survey paper list in the same research area (your survey)

[1] [Butler W. Lampson](https://dblp.org/pid/l/ButlerWLampson.html), [Martín Abadi](https://dblp.org/pid/a/MartinAbadi.html), [Michael Burrows](https://dblp.org/pid/73/615.html), [Edward Wobber](https://dblp.org/pid/88/1825.html):  
**Authentication in Distributed Systems: Theory and Practice.** [SOSP 1991](https://dblp.org/db/conf/sosp/sosp91.html#LampsonABW91): 165-182

[2] [Igor Konnov](https://dblp.org/pid/00/1088.html), [Jure Kukovec](https://dblp.org/pid/219/2203.html), [Thanh-Hai Tran](https://dblp.org/pid/141/9806.html):  
**TLA+ model checking made symbolic.** [Proc. ACM Program. Lang. 3(OOPSLA)](https://dblp.org/db/journals/pacmpl/pacmpl3.html#0001KT19): 123:1-123:30 (2019)