Security protocols

Strand Spaces, Scyther

SECURITY PROTOCOL ANALYSIS

- a **protocol** – set of rules that determine the exchange of messages between two or more principals
- **security protocols** – use cryptographic mechanisms to achieve security objectives
  (entity or message authentication, key establishment, ensuring secrecy, integrity, non-repudiation…)
- to prove –
  - complexity theoretic analysis – reduce the problem of attacking the protocol to some computationally hard problem
  - probabilistic reasoning (Scedrov …)
  - use of formal models with idealized cryptographic primitives (perfect cryptography) – we can use automatic tools (model checkers, theorem provers)

FORMAL MODELS FOR SECURITY PROTOCOL ANALYSIS

- **messages** are freely generated from Names of principals, Nonces and Keys, using the operators Concatenation and Encryption with a key
- principals know their private keys and public keys of others, can generate nonces
- a **role** is a procedure specified for each party in a protocol and that can be instantiated by any principal playing in the role
- perfect cryptography
- asynchronous communication
- formal model of **intruder**

INTRUDER BY DOLEV-YAO (1983)

intruder can

- intercept and read all messages
- decompose messages into their parts
- build and send new messages
- perform cryptographic operations but cannot break cryptography
- he might even be one of the principals running the protocol (corrupt principals)

assume an arbitrary number of principals

assume an arbitrary number of protocol executions

assume that protocol executions may be interleaved

Attacker model (Dolev-Yao)

- The attacker controls the communication medium. He can:
  - eavesdrop (and learn) on all messages
  - redirect messages
  - inject messages
  - apply cryptographic operations to the data he has learned
  - generated keys and random numbers

  - be a legitimate protocol participant (an insider), or an external party (an outsider), or a combination of both
  - start any number of parallel protocol runs between any principals including different runs involving the same principals and with principals taking the same or different protocol roles

FORMAL METHODS WITH AUTOMATIC TOOLS

- bounded number of sessions
- bounded number of principals
- each role has a bounded number of steps
- bounded message size

Most common approaches

- Discrete state-based analysis
- Logics of knowledge and belief

DISCRETE STATE-BASED ANALYSIS TOOLS

- Interrogator (Millen, 1984)
- NRL Protocol analyzer (Meadows, 1989)

automated intruder (Dolev-Yao), symbolic representation of states, use of lemmas to reduce infinite state space to finite one

- FDR checker (Lowe, 1996) – analysis of Needham-Schroeder public key protocol (based on CSP)
- MurΦ – analyze SSL
- project AVISPA
- ProVerif, Scyther, Tamarin

lower degree of abstraction, multiple protocols…

harder to use, state space explosion, semi-automated

LOGICS OF KNOWLEDGE AND BELIEF

- assumptions and protocol steps are translated into terms of the logic (idealisation).
- attempt to derive the cryptographic goals by application of the rules
- all principals are honest !?

BAN logic (Burrows, Abadi, Needham, 1989)

GNY logic (1990), AUTLOG

theorem proving tools based on Prolog

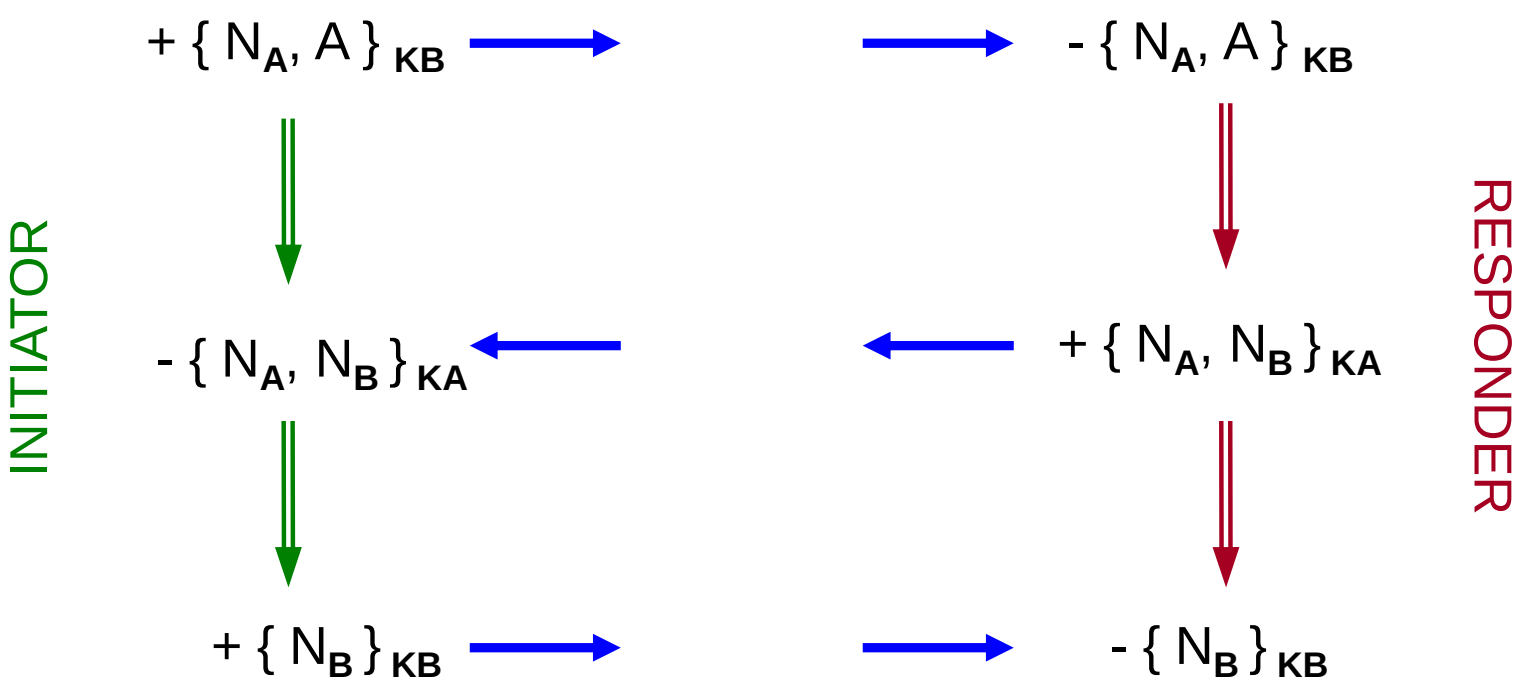
Spear, AAPA, Isabelle

deductive, abductive, mixed method of derivations

STRAND SPACE MODEL

- Thayer, Herzog, Guttman, 1999
- graph theoretic model of causal dependences of events in a cryptographic protocol (based on Dolev-Yao model)
- **strand** is a sequence of events, that represents either a protocol execution by a honest principal or a sequence of actions taken by the penetrator
- strand element – **node** – is either message transmission (positive term) or message reception (negative term)
- relation $\Rightarrow$ is used for elements, followed successively in the same strand
- relation $\rightarrow$ connects transmission (positive) node and reception (negative) node with the same term

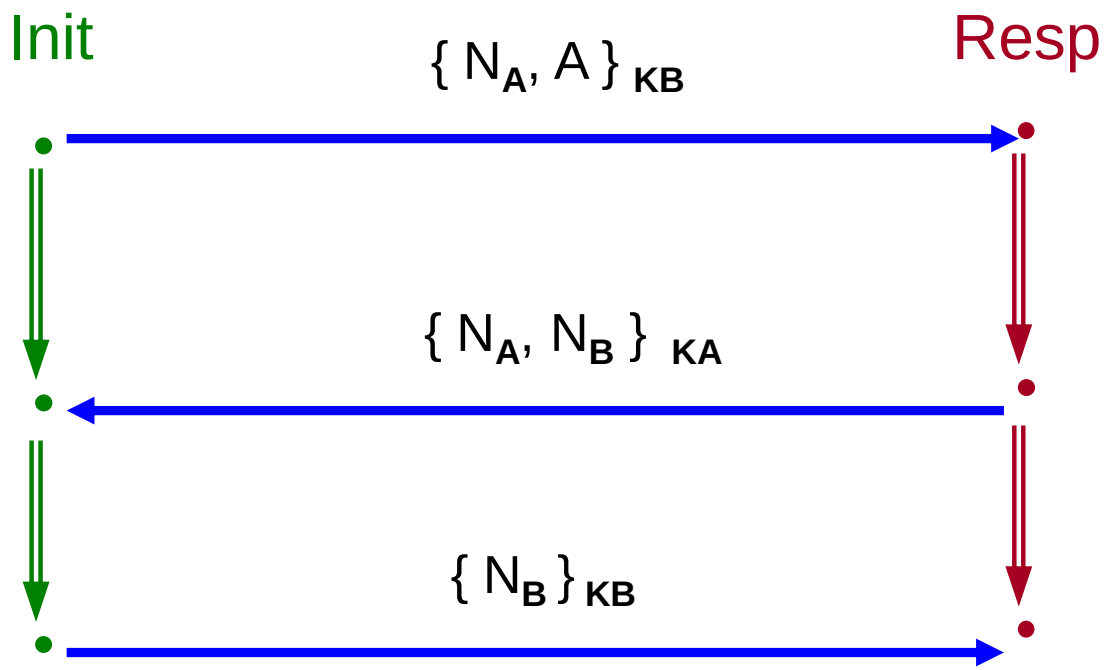
TWO TYPES OF HONEST STRANDS



STRAND SPACE MODEL

- **strand space** - collection of strands
- **bundle** – finite acyclic graph of nodes and edges representing causally well-founded execution
  - for every reception **- t** there is a unique transmission **+ t** where **+ t** $\rightarrow$ **- t**
  - when nodes **n** $\Rightarrow$ **m** on the same strand, then if **m** is in bundle, then also **n** is in bundle

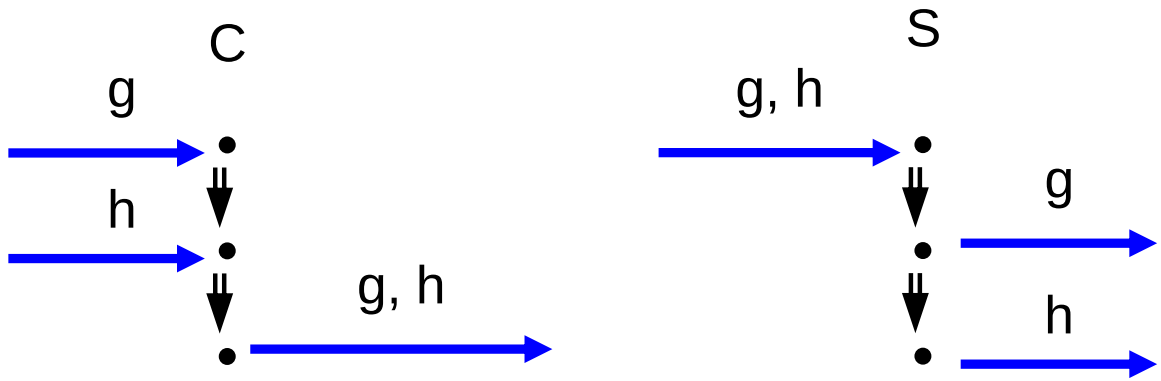
A BUNDLE FROM NEEDHAM SCHROEDER PROTOCOL



PENETRATOR STRANDS

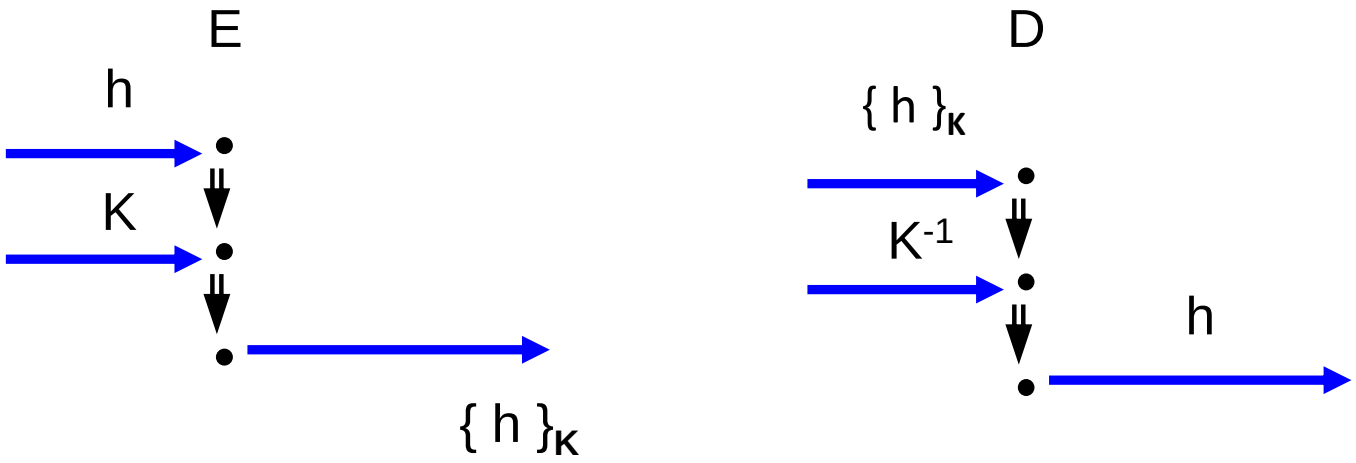
- Initiating values (message, key)

- Concatenate, Separate



PENETRATOR STRANDS

- Encrypt, Decrypt



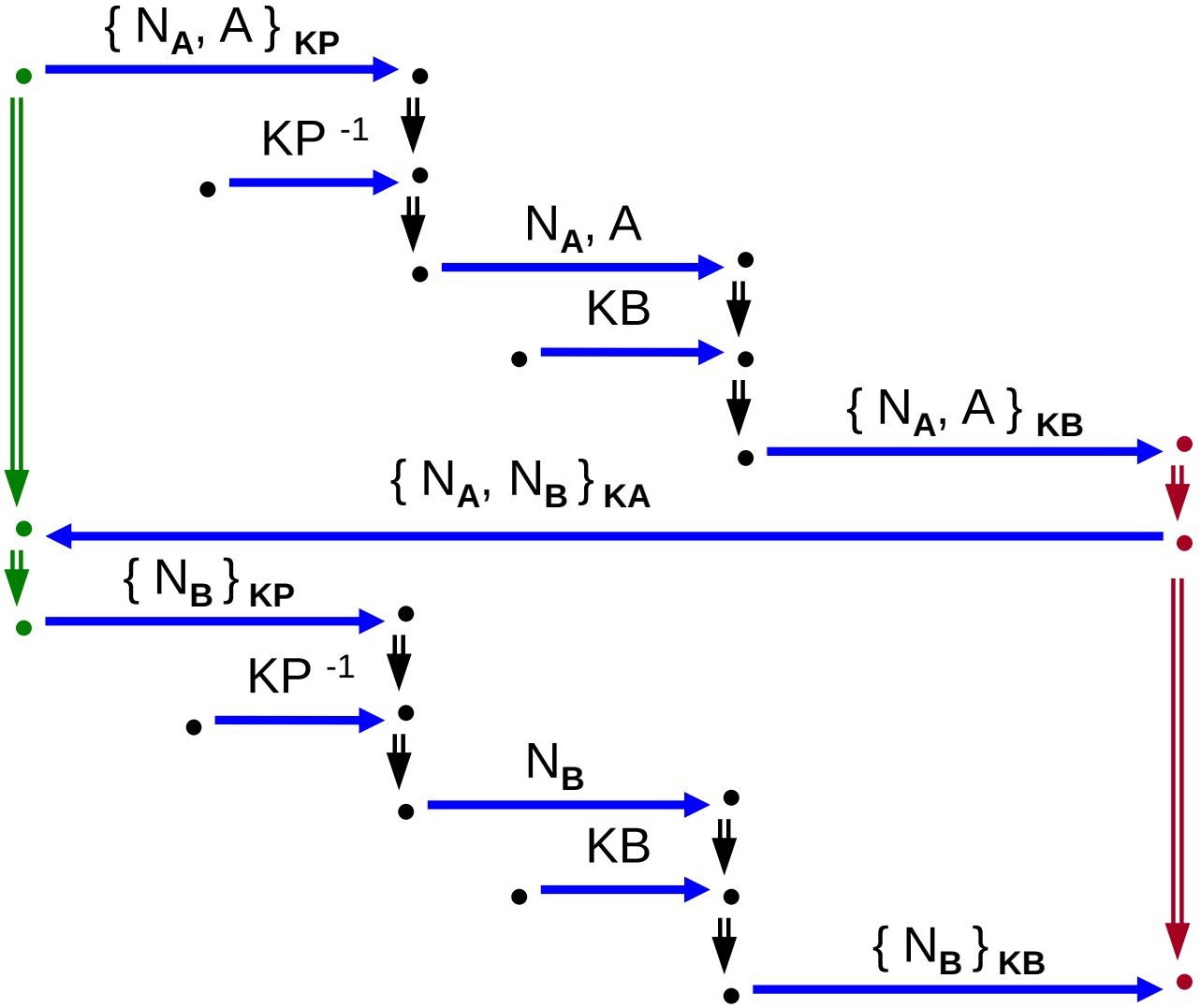
AN AUTHENTICATION GOAL

- suppose
  - bundle C contains a strand Resp[A,B,N_A,N_B]
  - N_B originates uniquely in C
  - KA^{-1} non-originating
  - $N_B \neq N_A$
- then is a strand Init[A,B,N_A,N_B] in C

AN SECRECY GOAL

- suppose
  - bundle C contains a strand Resp[A,B,N_A,N_B]
  - N_B originates uniquely in C
  - KA^{-1}, KB^{-1} non-originating
- then there is no node in C with term N_B

NSPK BUNDLE WITH LOWE ATACK



AUTOMATIC SECURITY PROTOCOL ANALYSIS USING STRAND SPACES

- finding paths in directed graphs – fast and effective algorithms
- on-line methods
- user-friendly interface

SCYTHER

Thank you for your attention