



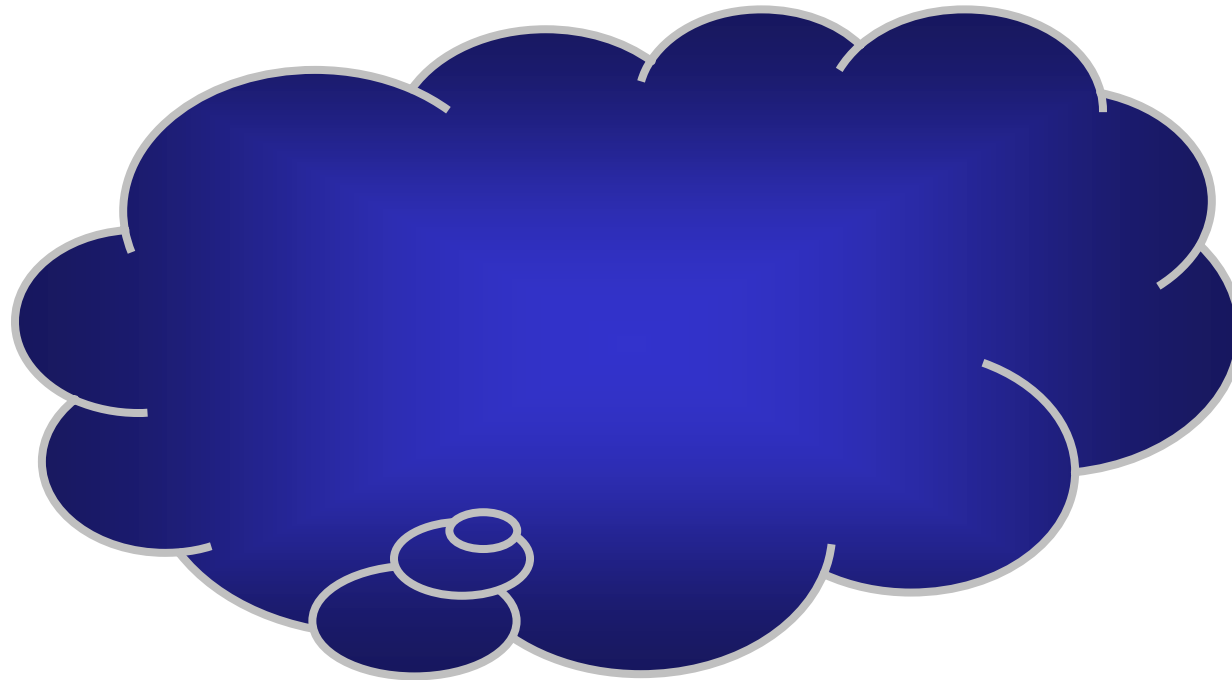
Protocol Design: Coming Down from the Cloud

Dieter Gollmann

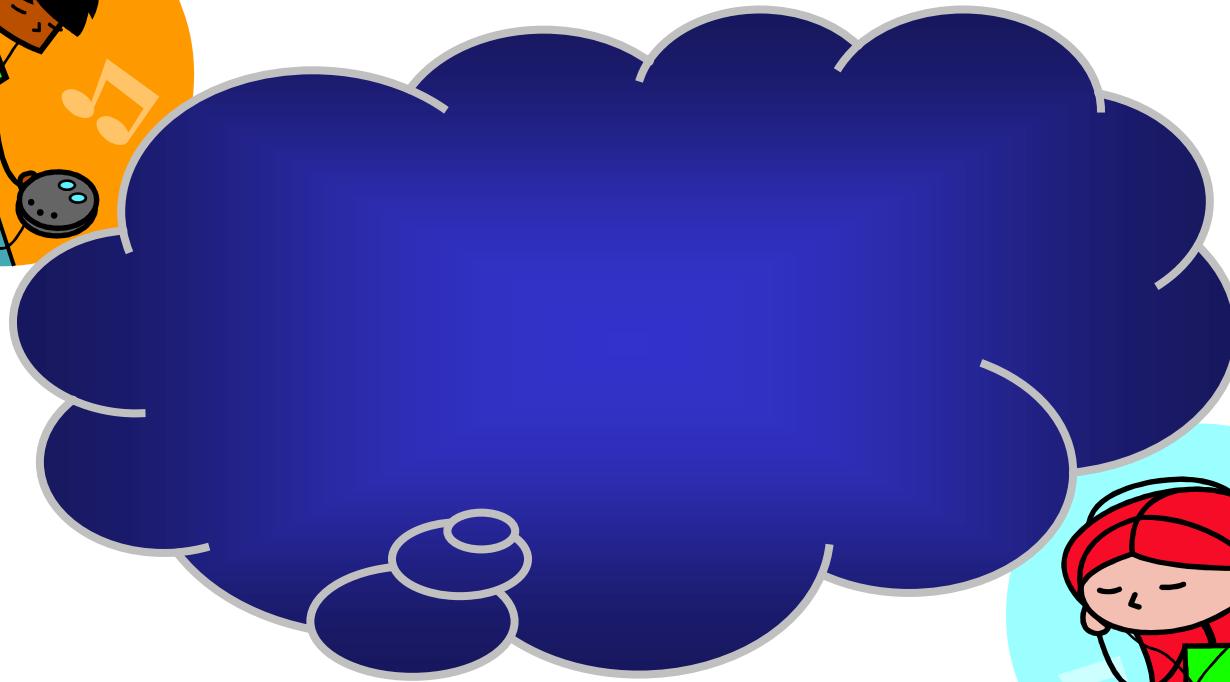
TU Hamburg-Harburg

diego@tuhh.de

The Cloud



The Cloud, Alice & Bob



Comment



§ It is customary to model the communications network as a cloud.

§ We do so when explaining protocols:

The Internet is a cloud.

Internal network structures are not considered.

§ We do so when analysing protocols:

The adversary is in control of all communications.

Internal network structures are not considered.

Comment



- § It is customary to call protocol participants Alice and Bob.
 - Each party has a single “identity”.
 - Internal structures of hosts (identities at different network layers) are not considered.
- § The metaphors we are using influence the way we think about security.
- § This talk is not specifically about RFID; it is about protocol design for novel applications.
- § RFID may facilitate novel applications; “old” security paradigms should not get in the way.

Moving the cloud



- § When the environment changes, established assumptions about security goals and security mechanisms, and even our language have to be adapted.
- § Protocol design & analysis have to consider the internal structure of the network.
- § Case studies:
 - Sensor networks: CANVAS
 - Mobile IPv6: binding updates

Before we start: a Fact



- § Assurances are typically required both that data actually come from its reputed source (data origin authentication), and that its state is unaltered (data integrity).
- § These issues cannot be separated – data which has been altered effectively has a new source; and if a source cannot be determined, then the question of alteration cannot be settled (without reference to a source).
- § Integrity mechanisms thus implicitly provide data origin authentication, and vice versa.

[Handbook of Applied Cryptography, p.359]

Case Study 1: Sensor Network



- § Based on work by Harald Vogt, ETH Zürich.
- § Network of sensor nodes.
- § Nodes do not use public-key cryptography.
- § Nodes do not know the identity of all nodes in the network.
- § Key setup: each node shares secret keys with direct neighbours and with nodes reachable via one intermediary.

Security Goals



- § Nodes can **create** new messages and **forward** messages.
- § Goal = “message authentication” (better: data integrity): forwarded messages cannot be manipulated or injected.
- § Defence against creation of bad messages is a separate issue.
- § MACs attached to messages: nodes can detect when a message is modified or has not come via the advertised nodes.

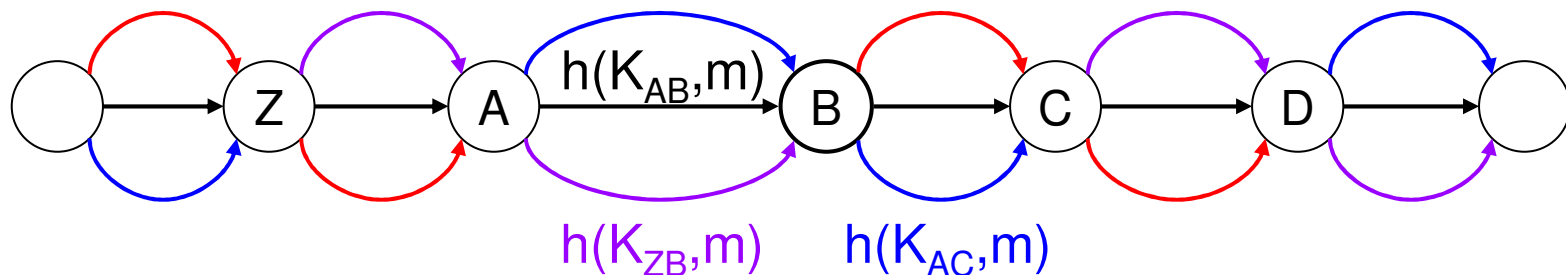
Canvas Protocol (H. Vogt)



§ B forwards message m received from A to C:

1. $A \rightarrow B$: m, Z, A, C, p, q, r
2. B verifies $p = h(K_{ZB}, m)$ and $q = h(K_{AB}, m)$
3. $B \rightarrow C$: $m, A, B, D, r, h(K_{BC}, m), h(K_{BD}, m)$

§ Similar protocol for creating new messages.



Security guarantees



- § Adversary: node trying to corrupt or inject **forwarded** messages; any **newly created** message is “legal”.
- § Adversary is **isolated** if no direct neighbour is also an adversary.
- § **Theorem: Canvas is robust against isolated adversaries.**
- § Formally verified with a protocol analysis tool.
- § **Canvas guarantees data integrity but does not provide data origin authentication!**

Integrity without authentication

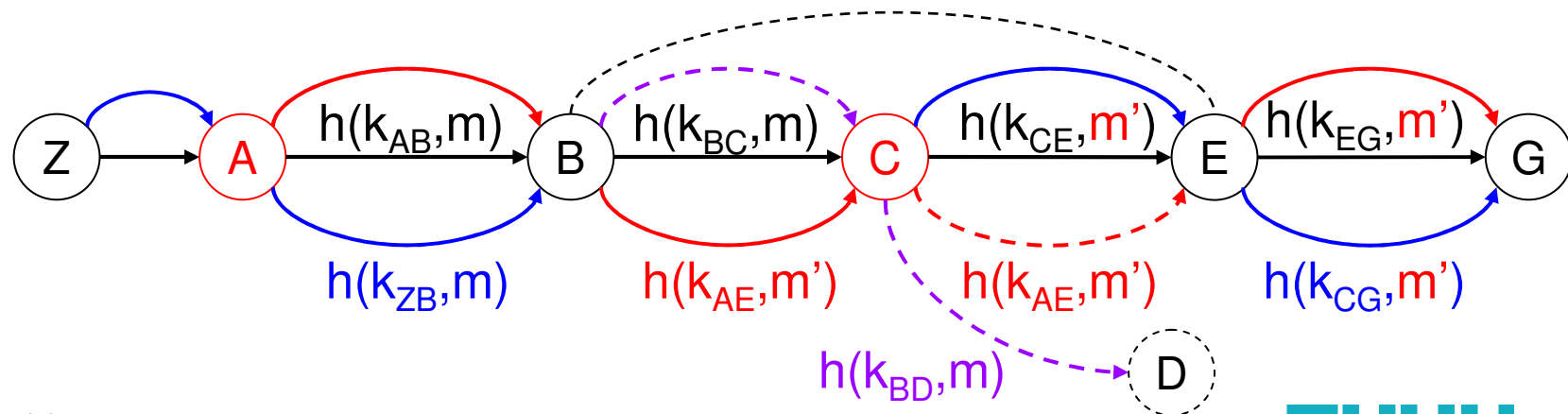


- § When identities of other nodes are unknown the sender's identity may not be an integral part of messages.
- § If we do not assume a completely insecure network, we may conclude that a message is received unchanged if a sufficient number of **independent witnesses** can vouch for this fact.
- § We can have data integrity without data origin authentication (outside the cloud).

Tearing the Canvas



- § To find an “attack” change the assumptions.
- § Adversaries **A** and **C** are isolated by **B**.
- § **A** and **C** have a common algorithm for modifying forwarded messages.
- § **A** knows **C**'s routing algorithm; inserts **m'**.



Remarks



- § Changing assumptions is a powerful attack method (for researchers).
- § It is not clear whether the “attack” presented is really a problem; this would depend on the actual application.
- § The attack could be prevented if nodes know more about the network structure.
- § This is against the spirit of the game: nodes would have to store more information.

Case Study 2: Mobile IPv6



- § A mobile node has a home address in its home network.
- § Messages sent to the home address are routed to the mobile node via a secure tunnel.
- § **Binding update** (performance optimisation): the mobile node informs a correspondent node about its current location.
- § The correspondent updates its **binding cache** to store the link between current location and home address.

Security Issues



If binding updates are unprotected,

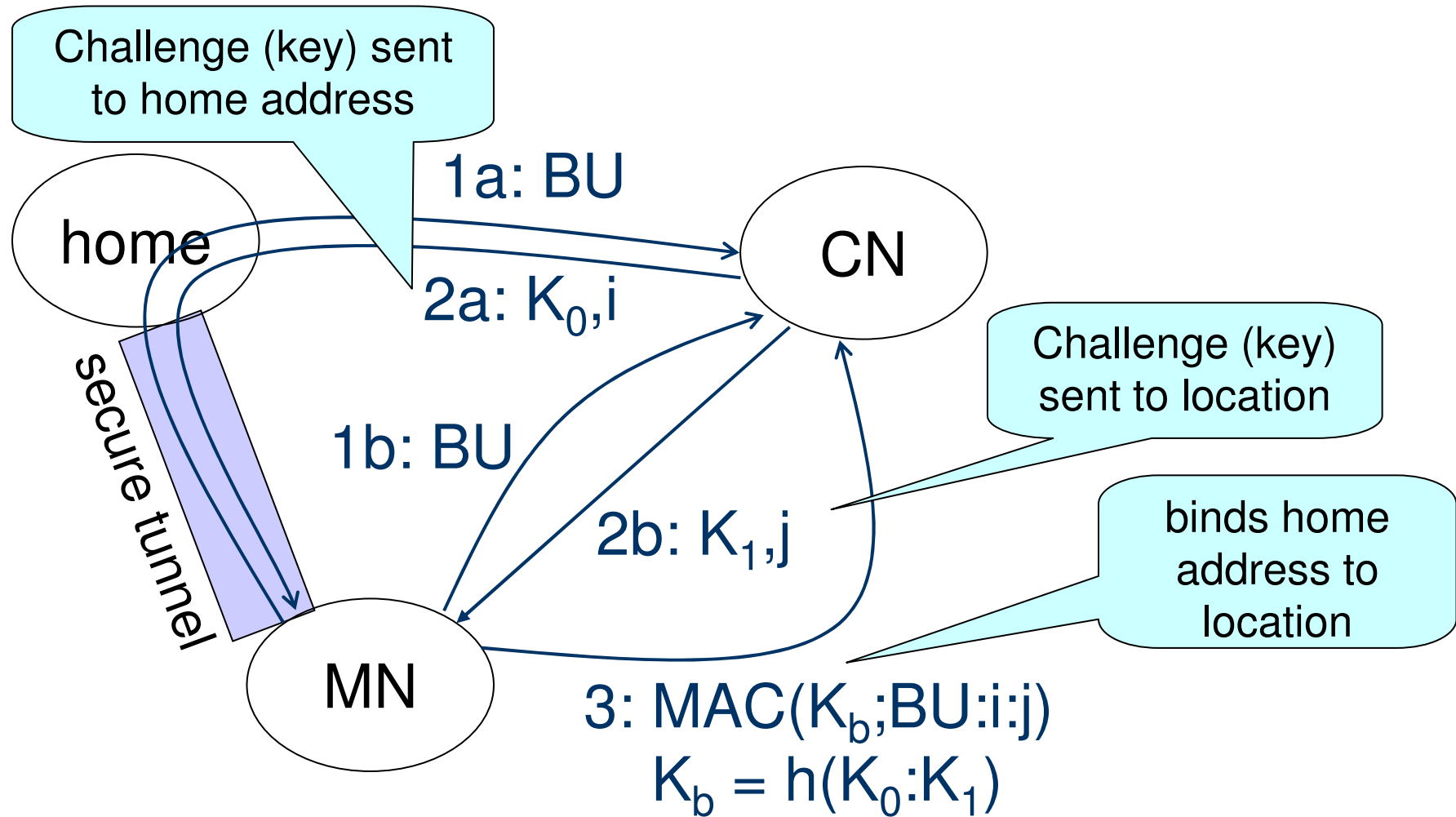
- § an attacker could pretend that another node has moved to its location to receive data intended for that node;
- § an attacker could pretend that another node has moved to a non-existing location so traffic to that node is lost (denial-of-service);
- § an attacker could pretend that it has moved to a location occupied by another node so that the victim is flooded with traffic requested by the attacker (bombing attack).

Secure Binding Updates



- § Unprotected binding updates would create havoc also with the fixed Internet; it is not possible to distinguish mobile IPv6 addresses from fixed addresses.
- § Data origin authentication is insufficient: the bombing attack would not be prevented.
- § The solution has to make the Internet only as (in)secure as it was without mobility.
- § RFC 3775: Mobility Support in IPv6.

Secure Binding Update



Remarks



- § Keys are sent in the clear, and could equally be interpreted as nonces.
- § Security based on **return routability**: correspondent checks that it receives a confirmation from the advertised location.
- § The protocol creates a binding between home address (identity in the home subnet) and current location (“location authentication”).
- § There is a growing number of protocols that bind identities at different layers: EAP, HIP, ...

Conclusions



If you are living in a cloud,

- § there are no structures in the network you could use to achieve security goals, (and security relies completely on the end systems)
- § you might make assumptions about the world that turn out to be wrong once you get a clear view,
- § you might not see problems that only become visible when you get a clear view on concrete applications.

Conclusions: Protocol analysis



- § Analysis in the “Dolev-Yao model” does not always give the most relevant results.
- § Model makes two independent assumptions:
 - The adversary can observe and manipulate all messages exchanged in a protocol run and can itself start protocol runs.
 - Cryptography is “perfect”: adversary only exploits algebraic properties of cryptographic operators and interactions between protocol messages.

Comment



§ The first assumption was already stated by Needham and Schroeder [1978]:

We assume that the intruder can interpose a computer in all communication paths, and thus can alter or copy parts of messages, replay messages, or emit false material. While this may seem an extreme view, it is the only safe one when designing authentication protocols.

Conclusions



- § Once, the world needed convincing that this general attack model made sense.
- § In the design of protocols for novel applications, we have to convince ourselves that more specific models make sense.
- § Protocol analysis methods should be able to capture those specific models.

Conclusions



Novel applications may introduce

- § novel threats and security requirements,
- § that can be addressed with novel types of security mechanisms,
- § while firmly established “truths” turn out to hold only in the general communications model we have grown accustomed to.



Thank you very much