A Time-Based Authentication Protocol for Link Layer Communications

SMU | Bobby B. Lyle School of Engineering

Ian Johnson, Daniel Engels, Jennifer Dworak

Problem Statement

Authentication is always a concern in network communications at any layer. Most authentication policies require the transmission of redundant or additional data alongside the body of a message. Because redundancy is often introduced at multiple layers, we present a link layer protocol to provide authentication using response time, at no cost with respect to data load.

Protocol Design

Our proposed protocol uses an induced delay on both ends of a communication to authenticate a sender. Before sending a message, a client induces an artificial delay. A recipient then validates that the correct delay was induced by measuring the response time. The delay is encoded using parallel keystreams that are generated on both sides of the communication using a pre-existing shared secret key and an initialization vector which the two parties compute using the Diffie-Hellman exchange.

Assumptions

Two key assumptions are made in the design of our protocol. First, it is assumed that each sender can send a message whenever they want (for example, in a CDMA network). Second, it is assumed that the propagation delay across the link is nearly constant.

Connection Setup

To perform connection setup, a 4-way handshake occurs. The first 3 messages consist of a Diffie-Hellman exchange, during which each party measures response time. The final message is a handoff message to the initiator.

Connection Teardown

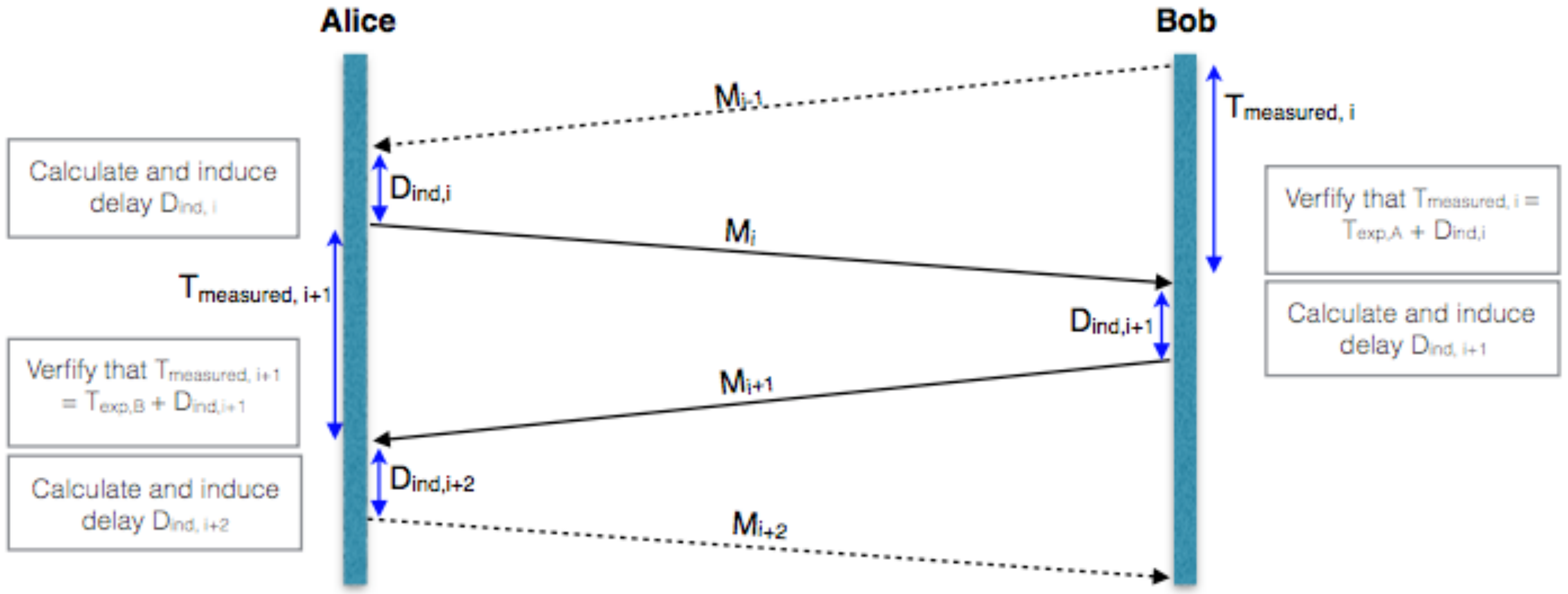
The teardown policy is a 3-way handshake. The initiator sends a teardown message, the recipient sends an ack and the initiator ends with a teardown-finish. Timeouts are in place to prevent hanging connections.

Computing Delay

Both parties share a secret key K and an initialization vector N_0. To compute each delay D_i, they extract M bits from the keystream KI created with a stream cipher with parameters K and N_0. D_i uses bits $Ki_{m(i)\ldots m(i+1)}$.

Standard Communication

Precondition: Alice and Bob share a secret key stream N generated with a shared secret key K



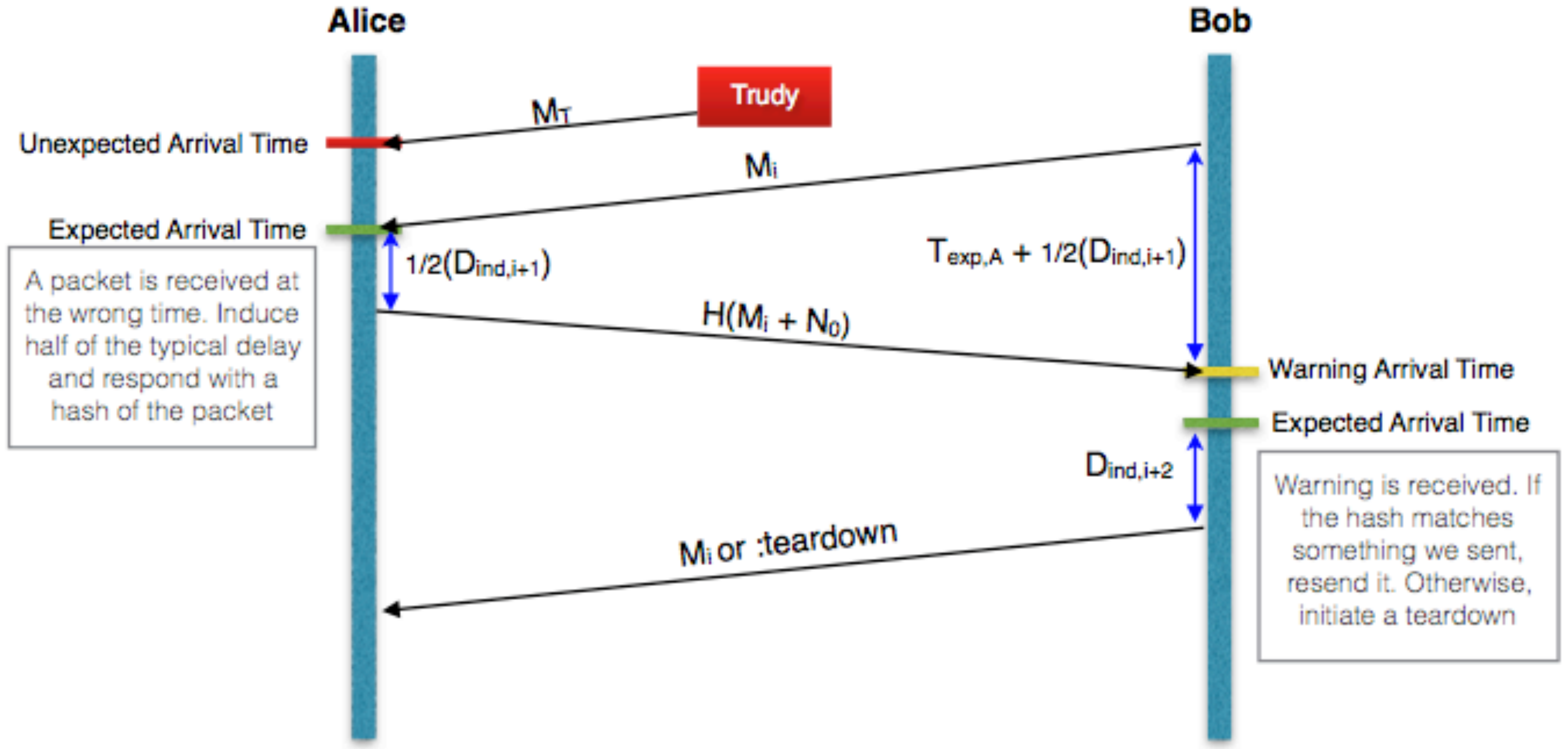
Calculate and induce delay $D_{ind, i}$

Verfify that $T_{measured, i} = T_{exp,A} + D_{ind,i}$

Calculate and induce delay $D_{ind, i+1}$

Verfify that $T_{measured, i+1} = T_{exp,B} + D_{ind,i+1}$

Calculate and induce delay $D_{ind, i+2}$



A packet is received at the wrong time. Induce half of the typical delay and respond with a hash of the packet

Warning is received. If the hash matches something we sent, resend it. Otherwise, initiate a teardown

Attempted Attack

When a malicious 3rd party (Trudy) attempts to send false information to Alice, she recognizes the threat and communicates it to Bob without alerting Trudy.