Bern University of Applied Sciences

Modeling a Bulletin Board Service based on Broadcast Channels with Memory

Severin Hauser and Rolf Haenni

Voting'18 @ FC'18, Curaçao, March 2, 2018

Outline

- Introduction
- Broadcast Channel With Memory
- Bulletin Board Service
- Conclusion

Outline

- Introduction
- Broadcast Channel With Memory
- Bulletin Board Service
- Conclusion

The Secure MPC Perspective

- The "voting problem" can be formulated as a secure multi-party computation (MPC) problem
  - Parties (voters) $P_1, \ldots, P_n$ with private inputs $x_i \in \{0, 1\}$
  - Common output $f(x_1, \ldots, x_n) = \sum_i x_i$
- Properties of secure MPC protocols
  - Privacy
  - Correctness
  - Independent inputs
  - Output delivery
  - Fairness
- Formal security definition based on ideal/real-model paradigm

Known Results from MPC Research

- Let $t \leq n$ be the number of corrupted parties
- For $t < \frac{n}{2}$
  - Secure MPC protocols exist for any function f
  - Even for computationally unbounded adversaries
  - Precondition: broadcast channel
- For $t \geq \frac{n}{2}$
  - Secure MPC protocols (without output delivery and fairness) exist for any function f
  - Only for polynomially bounded adversaries
  - Precondition: broadcast channel

Cryptographic Voting Protocols

- General MPC protocols are not applicable to real-world elections
  - Inefficient for large electorate
  - Limited connectivity of voters (vote-and-go)
  - No broadcast channel among voters
- Therefore, e-voting research focuses on designing specialized cryptographic voting protocols with additional parties such as
  - Election administration
  - Voting server
  - Independent authorities (of which at least some are honest) for tasks such as mixing or threshold decryption
  - Verifiers

  which communicate over point-to-point channels

Verifying an Election

- A precondition for verifying an election is a consistent view of the "election data"
- This is a Byzantine agreement problem, which can be solved using reliable broadcast protocols
- Same problems as general MPC approach
  - Inefficient for large number of parties
  - Limited connectivity of parties during protocol execution
- Therefore, several voting protocols in the literature refer to a broadcast channel with memory (BCM)
  - Memorization of all submitted messages
  - Delayed delivery
- Problem: no formal definition in literature

Outline

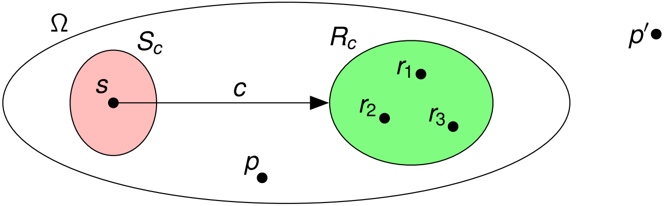
- Introduction
- Broadcast Channel With Memory
- Bulletin Board Service
- Conclusion

Ideal Channel

- A distributed system $\mathcal{D} = (\Omega, \Gamma)$ consist of:
  - Set of parties $\Omega = \{p_1, \ldots, p_n\}$
  - Set of ideal channels $\Gamma = \{c_1, \ldots, c_m\}$
- Ideal means:
  - instantaneous transmission
  - unlimited capacity
  - noiseless
  - total message ordering
- Every channel $c \in \Gamma$ defines:
  - Sender domain $S_c \subseteq \Omega$
  - Receiver domain $R_c \subseteq \Omega$
  - Message domain $\mathcal{M}_c$

Message Transmission

- If $s \in S_c$ transmits $m \in \mathcal{M}_c$ over c to R_c by calling
$$s : Send_c(m)$$
then every $r_i \in R_c$ receives m instantaneously
- Parties $p \in \Omega \setminus R_c$ can observe the transmission of m over c, but do not learn anything about m (except possibly its length)
- Parties $p' \notin \Omega$ can not even observe the transmission of m



Special Cases

	Ω	S_c	R_c
Broadcast channel	–	–	Ω
Public channel	–	Ω	–
Public broadcast channel	–	Ω	Ω
Authentic channel	–	$\{s\}$	–
Authentic broadcast channel	–	$\{s\}$	Ω
Confidential channel	–	–	$\{r\}$
Secure channel	–	$\{s\}$	$\{r\}$
Untappable channel	$\{s, r\}$	$\{s\}$	$\{r\}$

Channel with Memory

- A channel with memory $c \in \Gamma$ keeps track of all messages
- $\mathbf{M}_c = \langle m_1, \ldots, m_t \rangle$ is called channel history of $c \in \Gamma$, i.e.

  $$\mathbf{M}_c \leftarrow \mathbf{M}_c || m$$

  is updated each time a message m is transmitted over c
- Sender $s \in S_c$ transmits $m \in \mathcal{M}_c$ over c to R_c by calling

  $$Send_c(m)$$

- At any time, receiver $r_i \in R_c$ obtains current $\mathbf{M}_c$ by calling

  $$\mathbf{M}_c \leftarrow Receive_c()$$

Broadcast Channel with Memory

- A channel with memory $c \in \Gamma$ is a called broadcast channel with memory (BCM), if $R_c = \Omega$
- In e-voting protocols, voters use public BCM ($S_c = \Omega$) and authorities use authentic BCM ($S_c = \{s\}$)
- If $C \subseteq \Omega$ denotes a collection of BCMs, then $\mathbf{M}_C$ denotes the joint channel history of all channels
- Verification in an e-voting protocol relies on checking the integrity/consistency/plausibility/validity of the data included in $\mathbf{M}_C$

Outline

- Introduction
- Broadcast Channel With Memory
- Bulletin Board Service
- Conclusion

Ideal BCMs in the Real World

- Ideal BCMs do not exist in the real world
  - Transmission is not instantaneous
  - Messages can be lost or altered during transmission
  - Capacity is limited
  - Ambiguous message ordering
  - Stateless (no memory)
- In the real-world, ideal BCMs can at best be approximated
- For example by a bulletin board service (BBS), which is responsible for tracking the board history **B**

Bulletin Board Service

- To eliminate a BCM $c \in \Gamma$ from a distributed system (Ω, Γ), some additional bulletin board parties Φ are added
$$\Omega' = \Omega \cup \Phi$$
- Their goal is to offer jointly a bulletin board service to all parties from Ω
- Additionally, Ψ contains the channels necessary for connecting the parties from Ω with the bulletin board parties from Φ
$$\Gamma' = (\Gamma \setminus \{c\}) \cup \Psi$$
- Actual protocol run on (Ω', Γ') instead of (Ω, Γ)

Functionality

- Instead of broadcasting m over c, pairs $p = (m, \alpha)$ is posted to the BBS
  - p is send to one or multiple bulletin board parties
  - α = meta-data (e.g. for authentication)
- In general, posts are processed in blocks $b = (\{p_1, \ldots, p_s\}, \beta)$
  - Board history update: $\mathbf{B} \leftarrow \mathbf{B} || b$
  - β = publication evidence (e.g. signed hash chain header)
- Block size s determines publication mode
  - Buffered publication (s is fixed)
  - Immediate publication ($s = 1$)
  - Periodical publication (block added after some time)

Properties

- Authentication: only parties from S_c can post messages
- Non-Discrimination: all parties from S_c can post messages
- Well-Formedness: only messages from $\mathcal{M}_c$ are accepted
- Ordering: correct message order in board history
- Uniqueness: list of recorded messages is unique
- Completeness: all recorded messsges are returned

Basic Roles

- The bulletin board parties may have different roles:
  - Collectors: receive the incoming messages
  - Disseminators: return the board history (upon request)
  - Broadcasters: broadcast information about board history (e.g. a signed hash chain header)
  - Associates: support the BBS in achieving its properties
- Some of the bulletin parties may act as trustees in the usual sense (a threshold number of honest trustees suffices for maintaining the service)

Special Cases

- Single-party BBS (e.g. Helios, UniVote, . . .)
- Multi-party BBS
  - Byzantine agreement protocols
  - Blockchain-based public ledger
- vVote system bulletin board
  - Collectors: peers
  - Disseminator: WBB
  - Broadcaster: publisher

Outline

- Introduction
- Broadcast Channel With Memory
- Bulletin Board Service
- Conclusion

Conclusion

- Formal definitions of broadcast channels
  - Authentic broadcast channel
  - Public broadcast channel
  - Broadcast channel with memory
- Informal definition of bulletin board service
  - Extension of general channel model
  - Covers several existing bulletin board implementations
  - Proposal of (informal) properties
- Work in progress . . .