Formal Analysis of Multi-Party Contract Signing

Rohit Chadha	Steve Kremer	Andre Scedrov
University of Sussex	Université Libre de Bruxelles	University of Pennsylvania

Digital contract signing

- Use digital signatures to sign a contract over a network
- Special instance of fair exchange protocols
- Important issue for electronic commerce
- Naive 2-party protocol example:

$$\mathsf{A} \longrightarrow \mathsf{B} : S_A(\mathsf{contract})$$
$$\mathsf{B} \longrightarrow \mathsf{A} : S_B(\mathsf{contract})$$

Digital contract signing

- Use digital signatures to sign a contract over a network
- Special instance of fair exchange protocols
- Important issue for electronic commerce
- Naive 2-party protocol example:

$$\mathsf{A} \longrightarrow \mathsf{B} : S_A(\mathsf{contract})$$
$$\mathsf{B} \longrightarrow \mathsf{A} : ✗$$

Digital contract signing

- Use digital signatures to sign a contract over a network
- Special instance of fair exchange protocols
- Important issue for electronic commerce
- Naive 2-party protocol example:

  $\mathsf{A} \longrightarrow \mathsf{B} : S_A(\mathsf{contract})$

  $\mathsf{B} \longrightarrow \mathsf{A} :$ ✗
- Bob may be malicious and not send his signature
- Asymmetry: someone must be the first to sign

Properties of contract signing

- Fairness
  - If A gets B’s signature, then B can get A’s signature and vice-versa
- Timeliness
  - A signer does not get stuck
- Advantage
  - A signer has an advantage if
    - it has a strategy to complete the exchange
    - and it has a strategy to abort the exchange
- Abuse-freeness (provable advantage)
  - A signer cannot prove to an external party that it has the power to choose the outcome

Evolution of contract signing

In 1980, Even & Yacobi showed that there is no fair, deterministic two-party contract signing protocol.

- Randomized protocols
- Trusted Party, T intervenes
  - Use trusted party as a delivery authority
  - May cause a bottleneck . . .
- Trusted Party intervenes only in case of problem (optimistic approach)
  - More complex, and more error-prone . . .

Related work: formal methods & optimistic protocols

- [Shmatikov, Mitchell, 2000]
  - ➤ model-checker Murφ
  - ➤ invariant checking
- [Chadha, Kanovich, Scedrov, 2001]
  - ➤ specification in MSR
  - ➤ inductive proofs
- [Kremer, Raskin, 2002]
  - ➤ model-checker MOCHA
  - ➤ ATL (temporal logic with game semantics)
- [Chadha, Mitchell, Scedrov, Shmatikov 2003]
  - ➤ Protocol independent results on advantage

$\Rightarrow$ Only 2-party protocols studied

Multi-party contract signing

- n signers want to sign a contract
- Properties for a honest signer must hold against any coalition of dishonest signers, i.e., against up to $n-1$ dishonest signers
- Each signer must receive the signature of all other signers (topology is a full graph)

Multi-party protocols

- Astonishingly few so far
- [Asokan, Baum-Waidner, Schunter, Waidner, T.R. 1998]
  Optimistic synchronous multi-party contract signing
- [Baum-Waidner, Waidner, T.R. 1998 & ICALP 2000]
  Optimistic asynchronous multi-party contract signing
- [Garay, MacKenzie, DISC 1999]
  Optimistic asynchronous multi-party contract signing
- [Baum-Waidner, 2001]
  Optimistic asynchronous multi-party contract signing with reduced number of rounds

Overview of our results

- BW protocol
  - no attack has been found
  - not a proof of security—we only verified the structure of the protocol
- GM protocol
  - anomaly concerning abuse-freeness
    - easy to fix
  - several attacks on fairness
    - no attack found when $n = 3$
    - for $n = 4$, different attacks against signers P_1, P_2 and P_3 (but not P_4)
    - need to completely rewrite the recovery protocol

Protocol model

- Signers are players
- 3 versions of player described using guarded commands
  - honest : follow the protocol
  - optimistic: honest, but prefers waiting for other signers
  - dishonest : may send messages out of order and continue the main protocol after contacting T
- Messages are immediately available for reading
- Only structural flaws are considered
  - no modelling of the cryptographic primitives
- MOCHA cannot handle parametric specifications
  - C++ programs for the protocols, that generate the MOCHA specification for a given number of signers

GM protocol

[Garay, MacKenzie, DISC 1999]

- Recursive description of the protocol
- Protocol for signer P_i depends on position i
- The protocol is divided into n levels
  - ➤ In each protocol level specific promises are used
  - ➤ Implemented using private contract signatures (convertible designated verifier signatures)
- i-level protocol is triggered when P_i receives 1-level promises from P_{i+1} through P_n
- At i-level, P_i to P_1 exchange i-level promises
  - ➤ Agree on contract with promises, not signatures
- P_i through P_1 close higher level protocols
- After the n^{th}-level, actual signatures are exchanged

The i-level protocol

P_i P_{i-1} $\ldots$ P_1

Distribute 1 level promises

$(i-1)$ level protocol

Collect $(i-1)$ level promises

Exchange (i) level promises

GM abort and resolve for P_i

P_i may contact T if it does not want to wait anymore

- To abort, P_i sends an abort request to T
- To resolve, P_i sends a resolve request to T
  In the request, P_i sends a promise from each signer
  - if $j > i$, P_i sends the maximum level promise received from P_j on m
  - if $j < i$, P_i sends the maximum level of promises received from each of the signers $P_{j'}$, with $j' < i$

GM protocol for T

- Each signer may contact T only once
- T replies with a resolved contract or an abort token
- T may overturn an abort, but never a resolve
- T maintains the following information for each contract to decide when to overturn an abort
  - validated: a boolean indicating whether the contract has been validated or not
  - S: the set of indices of parties that have aborted
  - F: set of indices of parties which help T to decide when to overturn an abort

An attack on fairness

- The first attack was discovered when we found an error in the "proof"
- Consider the protocol instance where $n = 4$
- Using $\textsc{Mocha}$, we show that fairness does not hold for a honest P_2
  There is a path such that
  - P_1, P_3 and P_4 have P_2's signature
  - P_2 does not obtain all other signatures
- Similar attacks can be shown against P_1 and P_3
- No attacks discovered for $n = 3$ signers

An attack on fairness (P_2)

- P_1, P_3 and P_4 collude against P_2
- P_3 aborts at the beginning
  - T adds P_3 to S
- P_1 resolves, but T responds with an abort
  - T adds P_1 to S and P_2 to F
- P_2 tries to recover, but as P_2 is in F, T responds with an abort
- P_4 resolves and T overturns the abort

An attack on fairness (3)

More generally the attack scenarios are as follows

- dishonest P_{k1} aborts but continues the protocol
- dishonest P_{k2} tries to recover but does not succeed
  - as a side-effect it adds one or several signers to the set F
- honest P_{k3} tries to recover but does not succeed
- dishonest P_{k4} recovers and overturns the abort

Correcting the GM protocol

- Major revisions required
  - Getting the decision to overturn abort correct
  - Recovery protocol and T's protocol changed
- Central idea in the revision
  - Abort overturned if and only if T infers that each signer that contacted it in the past has been dishonest
  - Idea borrowed from Baum-Waidner protocol
- Mocha did not discover any attacks for both 3 and 4 signers

Conclusions

- First formal analysis of multi-party contract signing protocols
- Using the model-checker MOCHA and the logic ATL instances of two protocols have been verified
- New attacks have been discovered in the GM protocol
  - Abuse-freeness broken using side information given by T: easy fi x
  - Fairness broken when $n > 3$: requires major changes
- Fixed GM protocol
  - the protocol for T has been completely rewritten
  - number of different recovery requests has been reduced
  - verifi cation with MOCHA did not detect any error
- Model optimistic players in multi-party protocols

New work

- Fairness as invariant checking
  - advantage of invariant checking: error trace provided
- Analysis of the protocol with $t < n - 1$ dishonest signers
  - fairness can also be broken in a way such that:
    - one of the honest signers is fooled
    - another honest signer obtains the signed contract
    - no dishonest signer receives the signed contract

Future work

- Correctness proofs for BW and the fixed GM protocol
  - Using theorem provers to carry out the proof
  - Specification language should be rich enough to specify the protocols for any n
- Extend the analysis to a more complete model
  - Dolev-Yao-like intruder
  - Parametric verification
- Study different topologies, e.g. ring topologies in fair exchange
- Extend general results on advantage, presented in [Chadha, Mitchell, Scedrov, Shmatikov 2003] to multiparty protocols

An attack on abuse-freeness

- Note that P_1 cannot abort
- Abort responses include the signers that have aborted
- If P_1 receives an abort from T, P_1 must have sent a resolve request
- Use T as an oracle:
  - T verifies all promises in a resolve request
  - By answering to P_1, provides evidence that all signers have started the protocol

Attack on abuse-freeness contd..

- Using MOCHA for $n = 3$, we show that abuse-freeness does not hold for an optimistic P_3: P_1 and P_2 have a strategy to reach a state where
  - P_1 has an abort reply, and
  - P_1 and P_2 have a strategy to obtain P_3's signature
  - P_1 and P_2 have a strategy to prevent P_3 from getting a contract
- Easy fix: make abort replies to different signers indistinguishable