MPRI – Course on Concurrency

Lecture 14

Application of probabilistic process calculi to security

Catuscia Palamidessi
LIX, Ecole Polytechnique
kostas@lix.polytechnique.fr
www.lix.polytechnique.fr/~catuscia

Page of the course:
http://mpri.master.univ-paris7.fr/C-2-3.html

Plan of the lecture

- Randomized protocols for security
- Focus on protection of identity (anonymity)
  - The dining cryptographers
    - Correctness of the protocol
    - Anonymity analysis
  - Crowds (a protocol for anonymous web surfing)

Anonymity: particular case of Privacy

- To prevent information from becoming known to unintended agents or entities
  - **Protection of private data** (credit card number, personal info etc.)
  - **Anonymity**: protection of identity of an user performing a certain action
  - **Unlinkability**: protection of link between information and user
  - **Unobservability**: impossibility to determine what the user is doing

More properties and details at www.freehaven.net/anonbib/cache/terminology.pdf

Anonymity

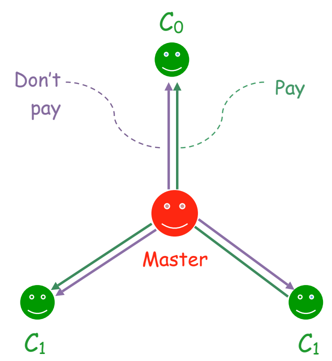
- Hide the identity of a user performing a given action
- The action itself might be revealed
- Many applications
  - Anonymous web-surfing
  - Anonymous posting on forums
  - Elections
  - Anonymous donation
- Protocols for anonymity often use randomization

The dining cryptographers

- A simple anonymity problem
- Introduced by Chaum in 1988
- Chaum proposed a solution satisfying the so-called "strong anonymity"

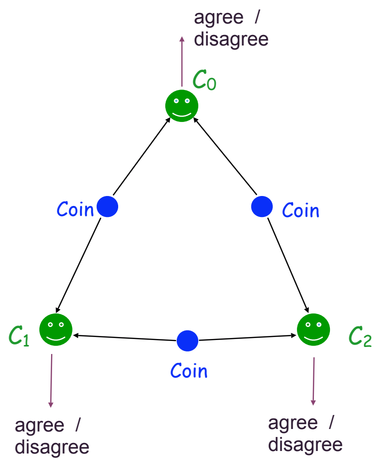
The problem

- Three cryptographers share a meal with a master
- In the end the master decides who pays
- It can be himself, or a cryptographer
- The master informs each cryptographer individually
- The cryptographers want to find out if
  - one of them pays, or
  - it is the master who pays
- **Anonymity requirement**: the identity of the paying cryptographer (if any) should not be revealed

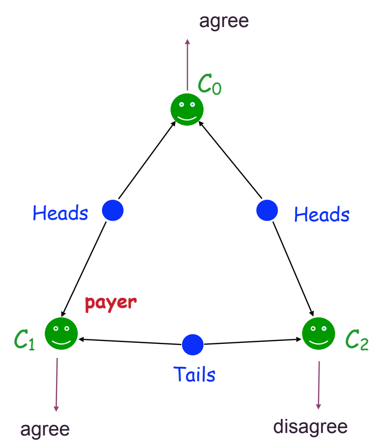


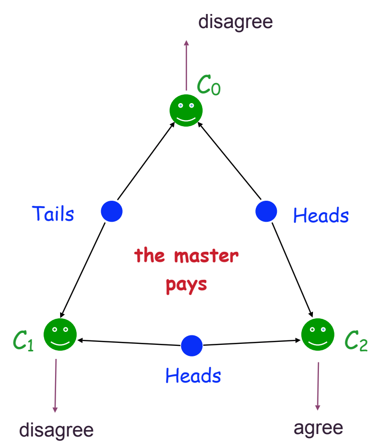
The protocol

- Each pair of adjacent cryptographers flips a coin
- Each cryptographer has access only to its adjacent coins
- Each cryptographer looks at the coins and declares **agree** if the coins have the same value and **disagree** otherwise
- If a cryptographer is the **payer** he will say the **opposite**
- Consider the **number of disagrees**:
  - **odd**: a cryptographer is paying
  - **even**: the master is paying



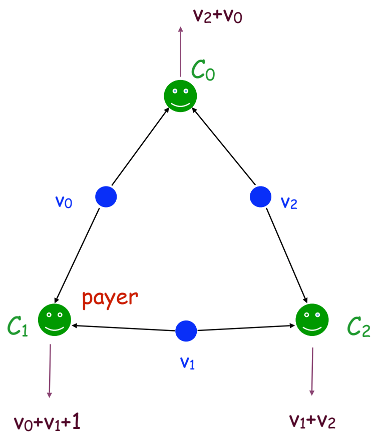
Examples





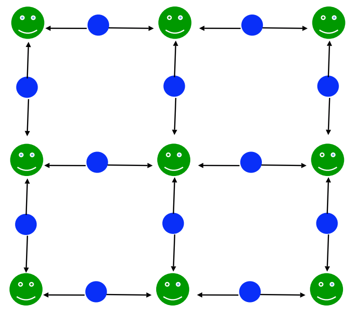
Correctness of the protocol

- Let $v_i \in \{0,1\}$ be the value of coin i
- Each cryptographer announces $v_{i-1}+v_i$ where + is the sum modulo 2:
  - 0 means agree
  - 1 means disagree
- The payer announces $v_{i-1}+v_i+1$
- The total sum is
  - $(v_0+v_1) + (v_1+v_2) + (v_2+v_0) = 0$ if the master pays
  - $(v_0+v_1+1) + (v_1+v_2) + (v_2+v_0) = 1$ if a cryptographer (C_1) pays



Correctness of the protocol

- The protocol is correct for any (connected) network graph
- The key idea is that all coins are added twice, so the cancel out
- Only the extra 1 added by the payer (if there is a payer) remains
- Note: this protocol could be extended to broadcast data anonymously, but the problem is that there in no distributed, efficient way to ensure that there is only one agent communicating the datum at each moment.

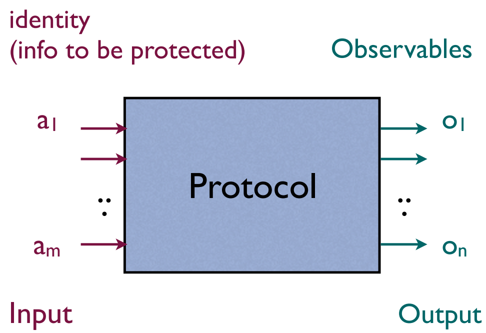


Anonymity of the protocol

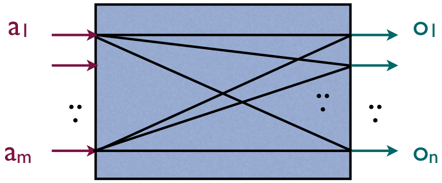
- How can we define the notion of anonymity?
- First we have to fix the notion of observable:
  - The anonymity property change depending on who is the observer / what actions he can see
    - An external observer can only see the declarations
    - One of the cryptographers can also see some of the coins

Notion of anonymity

Once we have fixed the observables, the protocol can be seen as a channel

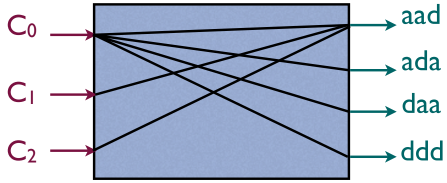


Notion of anonymity



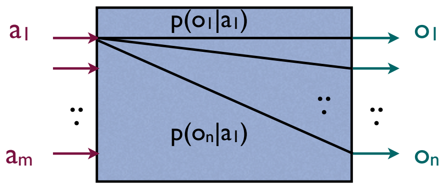
Protocols are noisy channels

Notion of anonymity



Example: The protocol of the dining cryptographers

Notion of anonymity



The conditional probabilities

Notion of anonymity

	o_1	...	o_n
a_1	$p(o_1\|a_1)$	...	$p(o_n\|a_1)$
:	:		
a_m	$p(o_1\|a_m)$		$p(o_n\|a_m)$

The conditional probabilities form a matrix.
In general the notion of anonymity will depend on these conditional probabilities

Notions of strong anonymity

In the following, a, a′ are hidden events, o is an observable

1. [Halpern and O'Neill - like] for all a, a′: p(a|o) = p(a′|o)
2. [Chaum], [Halpern and O'Neill]: for all a, o: p(a|o) = p(a)
3. [Bhargava and Palamidessi]: for all a, a′, o: p(o|a) = p(o|a′)

- (2) and (3) are equivalent. Exercise: prove it
- (1) is equivalent to (2),(3) plus p(a) = p(a′) for all a, a′
- the condition for all a, a′ p(a) = p(a′) depends on the input's distribution rather than on the features of the protocol

Anonymity in the Dining Cryptographers

- For an external observer the only observable actions are sequences of agree/disagree (daa, ada, aad, ...)
- Strong anonymity: different payers produce the observables with equal probability

p(daa | C_0 pays) = p(daa | C_1 pays)
p(daa | C_0 pays) = p(daa | C_2 pays)
p(ada | C_0 pays) = p(ada | C_1 pays)
...

- This is equivalent to requiring that

p(C_i pays) = p(C_i pays | $o_0o_1o_2$)

Expressing the protocol in probabilistic (value passing) CCS

Advantage: use model checker of probabilistic CCS to compute the conditional probabilities automatically

$$\begin{aligned} Master &= \textstyle\bigoplus_{i=0}^{2} p_i\ \bar{m}_i\langle 1\rangle.\bar{m}_{i+1}\langle 0\rangle.\bar{m}_{i+2}\langle 0\rangle.\mathbf{0} \\ &\quad \oplus p_m\ \bar{m}_0\langle 0\rangle.\bar{m}_1\langle 0\rangle.\bar{m}_2\langle 0\rangle.\mathbf{0} \\ Crypt_i &= m_i(x).c_{i,i}(y).c_{i,i+1}(z).\overline{out}\langle x+y+z\rangle.\mathbf{0} \\ Coin_i &= p_h\ \bar{c}_{i,i}\langle 0\rangle.\bar{c}_{i-1,i}\langle 0\rangle.\mathbf{0}\ \oplus\ p_t\ \bar{c}_{i,i}\langle 1\rangle.\bar{c}_{i-1,i}\langle 1\rangle.\mathbf{0} \\ DC &= (\nu\vec{m})(Master \mid (\nu\vec{c})(\Pi_{i=0}^{2} Crypt_i \mid \Pi_{i=0}^{2} Coin_i)) \end{aligned}$$

Expressing the protocol in probabilistic (value-passing) CCS

Anonymous action

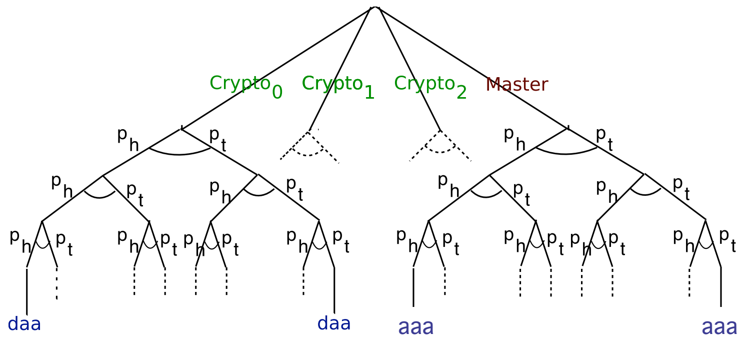
Observables

$$\begin{aligned} Master &= \textstyle\bigoplus_{i=0}^{2} p_i\ \bar{m}_i\langle 1\rangle.\bar{m}_{i+1}\langle 0\rangle.\bar{m}_{i+2}\langle 0\rangle.\mathbf{0} \\ &\quad \oplus p_m\ \bar{m}_0\langle 0\rangle.\bar{m}_1\langle 0\rangle.\bar{m}_2\langle 0\rangle.\mathbf{0} \\ Crypt_i &= m_i(x).c_{i,i}(y).c_{i,i+1}(z).\overline{out}\langle x+y+z\rangle.\mathbf{0} \\ Coin_i &= p_h\ \bar{c}_{i,i}\langle 0\rangle.\bar{c}_{i-1,i}\langle 0\rangle.\mathbf{0}\ \oplus\ p_t\ \bar{c}_{i,i}\langle 1\rangle.\bar{c}_{i-1,i}\langle 1\rangle.\mathbf{0} \\ DC &= (\nu\vec{m})(Master \mid (\nu\vec{c})(\Pi_{i=0}^{2} Crypt_i \mid \Pi_{i=0}^{2} Coin_i)) \end{aligned}$$

Probabilistic choices

Probabilistic automaton associated to the probabilistic π program for the D.C.

Assume we already fixed the scheduler



Anonymity of the protocol

- Assuming fair coins, we compute these probabilities

	daa	ada	aad	ddd
C_0	1/4	1/4	1/4	1/4
C_1	1/4	1/4	1/4	1/4
C_2	1/4	1/4	1/4	1/4

- Strong anonymity is satisfied

Anonymity of the protocol

- If the coins are unfair this is no longer true
- For example, if p(heads) = 0.7

	daa	ada	aad	ddd
C_0	0.37	0.21	0.21	0.21
C_1	0.21	0.37	0.21	0.21
C_2	0.21	0.21	0.37	0.21

- Now if we see daa, we know that c1 is more likely to be the payer

Anonymity of the protocol

- Even if we don't know the fact that the coins are unfair, we could find out using statistical analysis
- Exercise: suppose we see almost all the time one of the following announcements

  ada aad daa

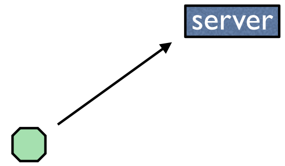
  - what can we infer about the coins?
  - then can we find the payer?
  - Now if we see daa, we know that C_0 is more likely to be the payer

Weaker notions of anonymity

- There are some problems in which it is practically impossible to achieve strong anonymity
- We need to define weaker notions
- In general, we need to give a quantitative characterization of the degree of protection provided by a protocol

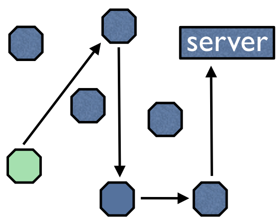
Example: Crowds

- A protocol for anonymous web surfing
- goal: send a request from a user (initiator) to a web serer
- problem: sending the message directly reveals the user's identity
- more efficient that the dining cryptographers: involves only a small fraction of the users in each execution



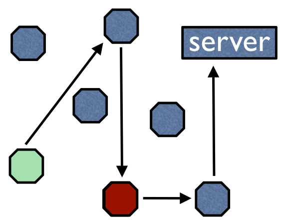
Crowds

- A "crowd" of n users participates in the protocol
- The initiator forwards the message to a randomly selected user (forwarder)
- A forwarder:
  - With probability p_f forwards again the message
  - With probability $1-p_f$ send the message directly to the server



Anonymity of the protocol

- Wrt the server: strong anonymity. The server sees only the last user
- More interesting case: some user is corrupted
- Information gathered by the corrupted user can be used to detect the initiator



Anonymity of the protocol

- In presence of corrupted users:
  - strong anonymity is no longer satisfied
  - A weaker notion called "probable innocence" can be achieved, defined as:

"the detected user is less likely to be the initiator than not to be the initiator"

Formally:

p(u is initiator | u is detected) < p(u is not initiator | u is detected)

Degree of protection: an Information-theoretic approach

- The entropy $H(A)$ measures the uncertainty about the anonymous events:

$$H(A) = -\sum_{a \in \mathcal{A}} p(a) \log p(a)$$

- The conditional entropy $H(A|O)$ measures the uncertainty about A after we know the value of O (after the execution of the protocol).
- The mutual information $I(A;O)$ measures how much uncertainty about A we lose by observing O:

$$I(A;O) = H(A) - H(A|O)$$

We can use (the converse of) the mutual information as a measure of the degree of protection of the protocol

Open problems

- Information protection is a very active field of research. There are many open problems. For instance:
  - Make model-checking more efficient for the computation of conditional probabilities
  - Active attackers: how does the model of protocol-as-channel change?
  - Inference of the input distribution from the observers

Bibliography

D. Chaum. The Dining Cryptographers Problem. *Journal of Cryptology*, 1, 65--75, 1988.

J.Y. Halpern and K. R. O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 13(3), 483-512, 2005

K. Chatzikokolakis, C. Palamidessi , P. Panangaden. Anonymity Protocols as Noisy Channels. *Information and Computation.* To appear. 2007. http://www.lix.polytechnique.fr/~catuscia/papers/Anonymity/Channels/full.pdf