Local perspective of mixing - a CSP approach

Stathis Stathakidis

Department of Computing
University of Surrey, UK

15 October 2012

1 Mixnets

2 Assumptions

3 Algorithms

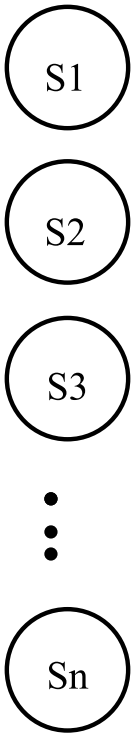
4 CSP

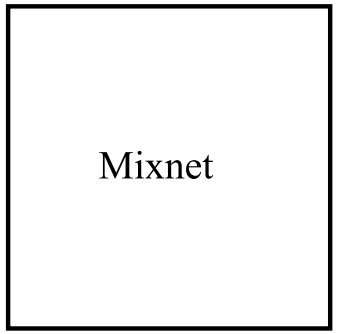
5 Problems and future work

6 Questions

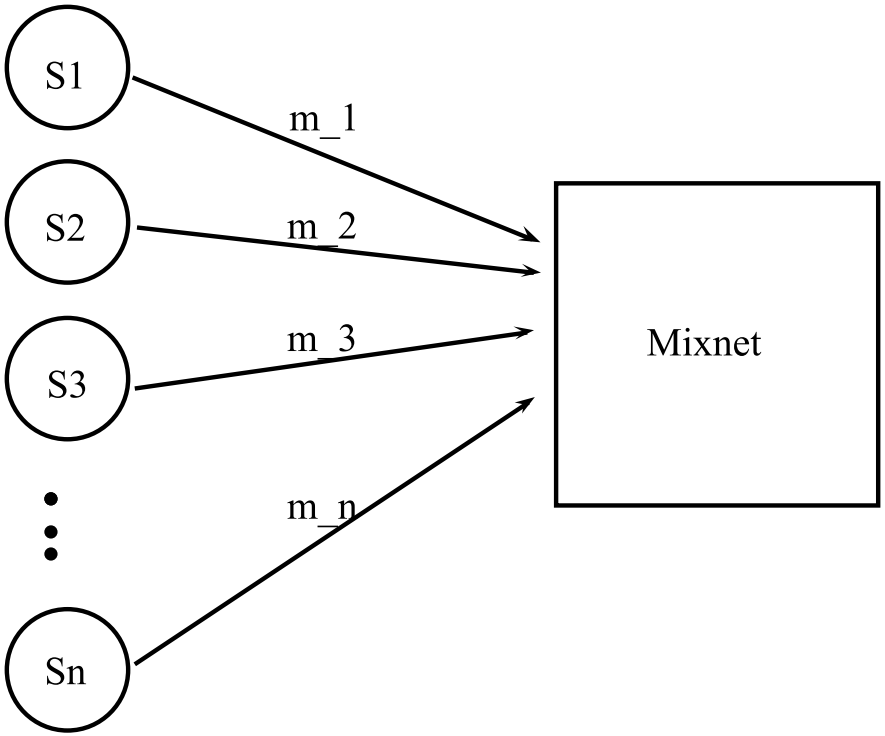
1 Mixnets
2 Assumptions
3 Algorithms
4 CSP
5 Problems and future work
6 Questions

- cryptographic protocol
- hides (unlink) the correspondence between its inputs and outputs
- consists of mix servers
- Chaum, 1981

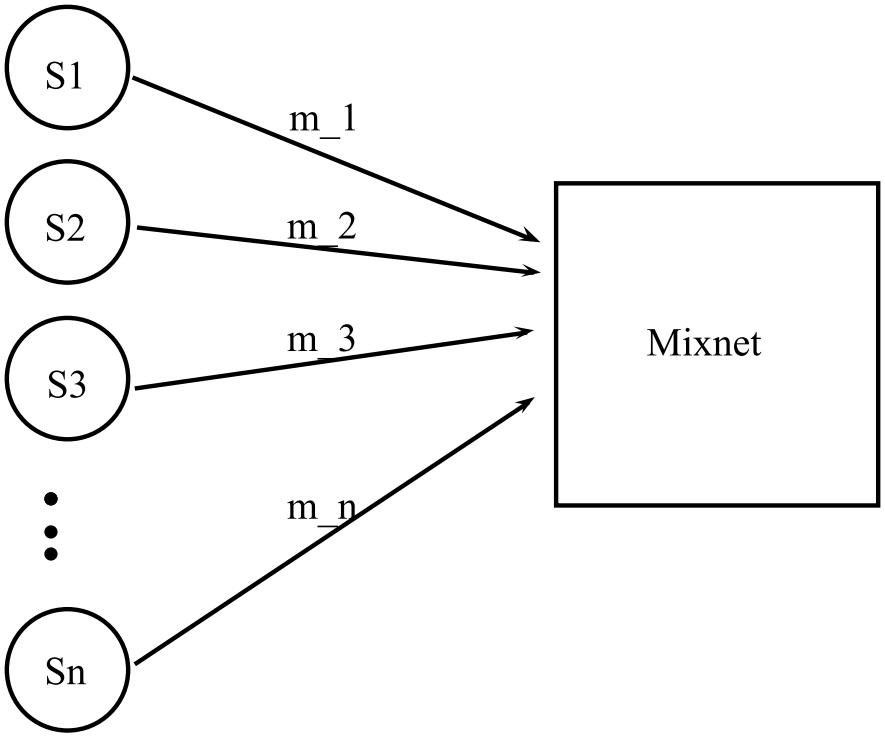




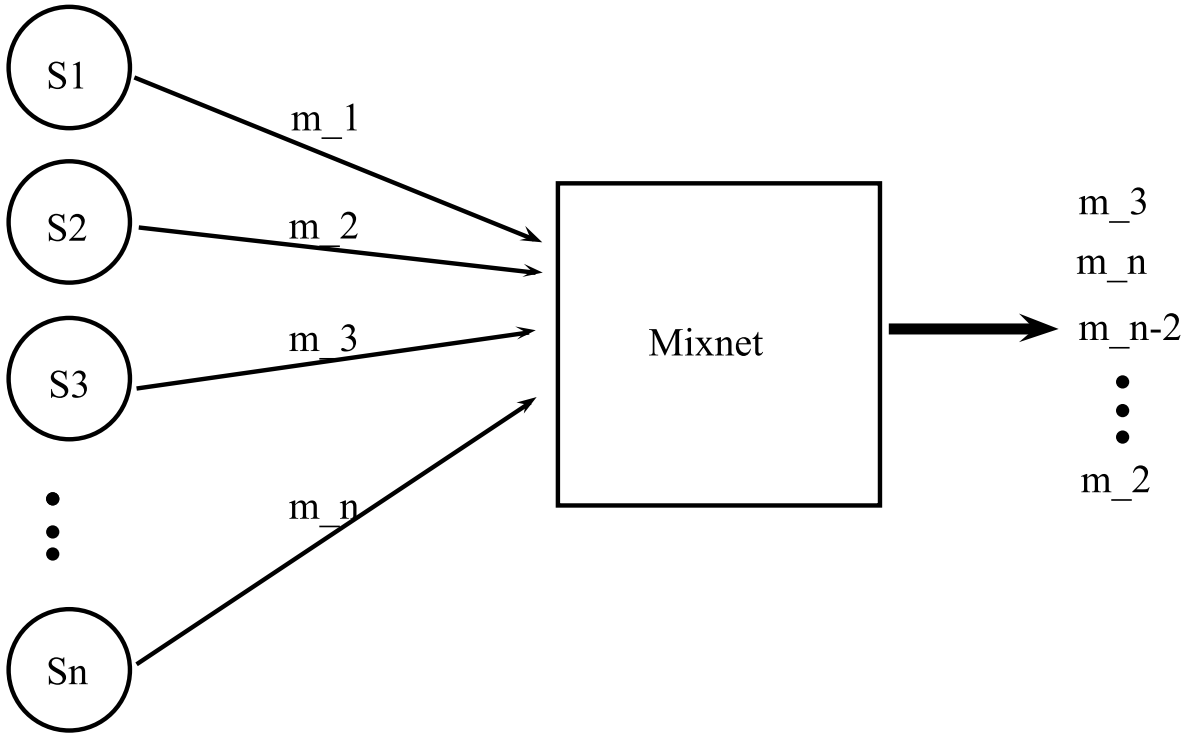
- Sender $Si, 1 < i < n$



- Sender $Si, 1 < i < n$
- Si sends m_i



- Sender $Si, 1 < i < n$
- Si sends m_i
- Mixnet operates



- Sender $Si, 1 < i < n$
- Si sends m_i
- Mixnet operates
- Outputs in random order

Uses

- RFID tags
- anonymous web browsing
- mainly in e-voting

- Prêt à Voter
- Helios
- Civitas
- . . .

Chaumian Mixnets

- untraceable mail system - 1981
- layers of encryptions - onion
- ciphertext proportional to the number of mix servers
- $c = E_{PK_1}(E_{PK_2} \dots E_{PK_{n-1}}(E_{PK_n}(m)\dots))$
- ciphertext c is delivered to the first mix server
- each mix server peels off the outer layer
- $m = D_{SK_n}(D_{SK_{n-1}} \dots D_{SK_2}(D_{SK_1}(c)\dots))$

Re-encryption Mixnets

- Park et al., 1993
- ciphertext's size irrelevant to the number of servers
- two variations
  - decryption at the end of the process (threshold)
  - partial decryption

Other

- parallel Mixnet
- hybrid Mixnets

So far

- global perspective of mixing
- when a server is found dishonest then it is either excluded or replaced
- no more information is given: how? who? when?
- a third party is involved - time consuming

- local perspective - how each mix server behaves
- output the final result without delay
- eliminate the existence of a third party

1 Mixnets
2 Assumptions
3 Algorithms
4 CSP
5 Problems and future work
6 Questions

Algorithm:

- unique
- unambiguous
- accurate
- run by each server

WBB

- secure and trusted
- public - anyone can read from it
- only servers can post on it
- communication channels are secure (read and post)
- gives accurate record of what is posted

Servers

- know their positions in the Mixnet
- same potential view to the WBB
- active during the process
- can perform the basic cryptographic operations

Other

- do not model the underlying cryptography
- proofs and ciphertexts as an entity
- no network traffic manipulation
- return proofs and verdicts in a timely fashion

1 Mixnets

2 Assumptions

3 Algorithms

4 CSP

5 Problems and future work

6 Questions

Algorithm 1 Mixserver

1: **if** $i == j$ **then**
2: $\quad$ *Mixing*(i, j)
3: **else**
4: $\quad$ *Checking*(i, j)
5: **end if**

Mixing(i,j)

- *i* looking for latest server *j* with “good” proofs
- *i* operates on *j*’s ciphertexts
- *i* posts its proofs and verdicts on the WBB
- *i* claims its proofs as “good”
- update the last server with “good” proofs

Algorithm 2 Mixing

1: **if** $i == 1$ **then**
2: $\quad$ *ReadProofs*$(i, lastGood, WBB)$
3: $\quad$ *Operate*(i)
4: $\quad$ *PostProofs*(i, P_i)
5: $\quad$ *PostVerdict*$(i, Verdict)$
6: $\quad$ *Mixserver*$(i, j + 1, lastGood + 1)$
7: **else**
8: $\quad$ *ReadProofs*$(i, lastGood, WBB)$
9: $\quad$ *Operate*(i)
10: $\quad$ *PostProofs*(i, P_i)
11: $\quad$ *PostVerdict*$(i, Verdict)$
12: $\quad$ *Mixserver*$(i, j + 1, i)$
13: **end if**

Checking(i,j)

- *i* reads *j*'s proofs from the WBB
- *i* posts its verdict about *j*'s proofs on the WBB (update)
- if the read proofs are "good" then $j++$
- update the last server *j* with "good" proofs
- else *j* is not considered as server with "good" proofs

Algorithm 3 Checking

1: *ReadProofs*(i, j, P_j, WBB)
2: *PostVerdict*$(i, j, Verdict)$
3: **if** $P_j == good$ **then**
4: *Mixserver*$(i, j + 1, j)$
5: **else**
6: *Mixserver*$(i, j + 1, lastGood)$
7: **end if**

- can do anything
- can refuse to read and produce proofs
- CHAOS in CSP - most non deterministic process
- too dishonest! is STOP enough?

- can do anything
- can refuse to read and produce proofs
- CHAOS in CSP - most non deterministic process
- too dishonest! is STOP enough? probably not

- accepts read and post queries
- anyone can read from it
- only servers can post on it
- initially consists of sequence of pending proofs
- and sequence of unknown verdicts

1 Mixnets

2 Assumptions

3 Algorithms

4 CSP

5 Problems and future work

6 Questions

- Communicating Sequential Processes
- Hoare, 1978
- tool for specifying and verifying concurrent systems
- subsystems which operate concurrently and interact each other
- need of a model checker - FDR

Datatypes

N: number of mix servers

nametype *NumberOfServers* = {1..N}

nametype *Servers* = NumberOfServers

datatype *Proofs* = good | bad | pending

Verdicts = {true, false}

Channels

operate: Servers

read_proofs: Servers.Servers.Proofs

read_verdicts: Servers.Servers.Verdicts

post_proofs: Servers.Proofs

post_verdicts: Servers.Servers.Verdicts

Mixserver's alphabet

$$
\begin{aligned}
alphaMIXSERVER(i) = \\
&\{read\_proofs.i.j.p, \\
&read\_verdicts.i.j.v, \\
&post\_proofs.i.p, \\
&post\_verdicts.i.j.check(p), \\
&operates.i \\
&\mid j \leftarrow Servers, p \leftarrow Proofs, v \leftarrow Verdicts\}
\end{aligned}
$$

Disjoint alphabet

DISHONEST

$$DISHONEST(i) = CHAOS_{\{alphaDISHONEST(i)\}}$$

Dishonest's alphabet

$$alphaDISHONEST(i) = \{read\_proofs.i.j.p, read\_verdicts.i.j.v, post\_proofs.i.p, post\_verdicts.i.j.check(p), operates.i \mid j \leftarrow Servers, p \leftarrow Proofs, v \leftarrow Verdicts\}$$

- $\Box$ the choice of what happens is in the hands of the environment
- $\Box$ the choice of what happens is in the hands of the process
- ? input
- ! output

neither an input nor an output can occur until the environment is willing to allow it

WBB

$$
\begin{aligned}
&WBB\ (Sqp, Sqv) \mathrel{\widehat{=}} \\
&\quad read\_proofs?i?j!nth(Sqp, j) \rightarrow WBB(Sqp, Sqv) \\
&\quad \Box \\
&\quad read\_verdicts?i?j!nth(Sqv, j) \rightarrow WBB(Sqp, Sqv) \\
&\quad \Box \\
&\quad post\_proofs?i?proof \rightarrow WBB(update(Sqp, i, proof), Sqv) \\
&\quad \Box \\
&\quad post\_verdicts?i?j?verdict \rightarrow WBB(Sqp, update(Sqv, j, verdict))
\end{aligned}
$$

WBB's alphabet

$$
\begin{aligned}
&alphaWBB = \\
&\{| \ read\_proofs, read\_verdicts, post\_proofs, post\_verdicts \ |\}
\end{aligned}
$$

Interleave means no explicit communication

GOOD SERVERS

$$GOOD\_SERVERS = |||_{i \in setOfHonestServers} Mixserver(i, 1, 0)$$

GOOD SERVER's alphabet

$$alphaGOOD = \cup(alphaMIXSERVER(i))$$

Interleave means no explicit communication

BAD SERVERS

$$BAD\_SERVERS = |||_{i \in setOfDishonestServers} DISHONEST(i)$$

BAD SERVER's alphabet

$$alphaBAD = \cup(alphaDISHONEST(i))$$

There is no direct communication between servers

We use interleaving instead of parallel

SERVERS

$$SERVERS = GOOD\_SERVERS \,|||\, BAD\_SERVERS$$

SERVERS's alphabet

$$alphaSERVERS = \cup(alphaGOOD, alphaBAD)$$

Interleaved processes do not synchronise on events even when their alphabets do overlap

SERVERS and WBB in parallel

Use of alphabetised parallel

SERVERS

$$SYSTEM = SERVERS \ {}_{alphaSERVERS}\|_{alphaWBB} \ WBB$$

SERVERS's alphabet

$$alphaSYSTEM = \cup(alphaSERVERS, alphaWBB)$$

SPEC

$SPEC = tally \rightarrow STOP$

Assertions

- checks to be carried out
- used to state properties which are believed to hold
- load in FDR alongside the processes
- deadlock and livelock freedom, traces, failures divergence etc

Checks

- hide the tally event
- deadlock free - never stop (robustness)
- livelock free
- divergence free - no unlimited sequences of internal action τ
- SYSTEM failures/divergences-refines SPEC

SPEC

$assertSPEC \sqsubseteq_{\mathrm{FD}} SYSTEM \setminus \{| tally |\}$

1 Mixnets
2 Assumptions
3 Algorithms
4 CSP
5 Problems and future work
6 Questions

- the servers run this algorithm at the same time
- no one can block the protocol from happening

- control process which prevents the WBB from reading pending proofs $\rightarrow$ time is involved
- more sanity checks

- ☺ split into proofs and ciphertexts
- ☺ add time restrictions - timed CSP
- ☹ drop the WBB - single point of failure - Verificatum uses no WBB - huge challenge
- ☺ distributed WBB? yes, but in CSP . . . ? ☹☹☹☹
- ☹ state space explosion - reduce it
- ☹ model the decryption

1 Mixnets

2 Assumptions

3 Algorithms

4 CSP

5 Problems and future work

6 Questions

Thank you!

Thank you!

Questions?