

Robustness Guarantees for Anonymity

Gilles Barthe¹ Alejandro Hevia² **Zhengqin Luo**³
Tamara Rezk³ Bogdan Warinschi⁴

¹IMDEA Software, Madrid, Spain

²Dept. of Computer Science, Universidad de Chile, Chile

³INRIA Sophia Antipolis-Méditerranée, France

⁴University of Bristol, United Kingdom

Cosyproof '10

To appear in CSF'10

How (or methodologies) to prove/verify protocols?

- . . . computational soundness, symbolic analysis, universal composition, faithful abstraction, certified game-based proof, typing, bisimulation, applied pi calculus . . .

How (or methodologies) to prove/verify protocols?

- . . . computational soundness, symbolic analysis, universal composition, faithful abstraction, certified game-based proof, typing, bisimulation, applied pi calculus . . .

What properties to prove?

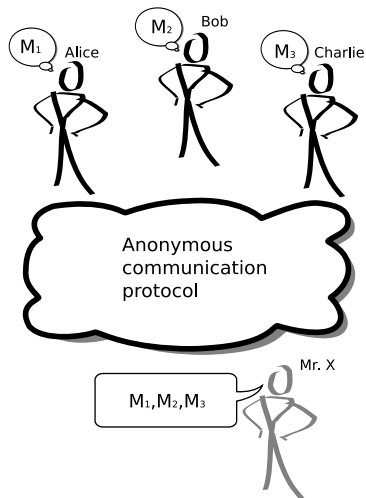
Anonymous Communication Protocols: What do they provide?

Definitions of anonymity [MH06]

... Receiver anonymity, Sender Unlinkability, Receiver Unlinkability, Sender-Receiver Unlinkability, Sender Anonymity, Strong Sender Anonymity, Receiver Anonymity, Strong Receiver Anonymity, ...

Useful in many contexts

- Election protocols
- Auction protocols
- Censorship resistance



Robustness: A cautionary tale

Robustness (message integrity) is not provided by anonymity.

Robustness: A cautionary tale

Cryptographers at Cosyproofs want to comment the hotel anonymously

- Alice: **Internet is too slow;**
- Bob: **Water is not hot enough;**
- Carol: **Desserts are too hard to cut;**

Robustness: A cautionary tale

They put envelopes inside a small box:

- Envelope: **Internet is too slow;**
- Envelope: **Water is not hot enough;**
- Envelope: **Desserts are too hard to cut;**

Robustness: A cautionary tale

Some guy (magically) replace all the envelops:

- Envelop: **Internet is not too slow;**
- Envelop: **Water is hot enough;**
- Envelop: **Desserts are delicate;**

Robustness: A cautionary tale (continued)

Facts

- Identities of message senders are hidden – anonymity is guaranteed;
- Contents of messages are not protected – robustness is **not guaranteed**.

Robustness: A cautionary tale (continued)

Facts

- Identities of message senders are hidden – anonymity is guaranteed;
- Contents of messages are not protected – robustness is **not guaranteed**.

Standard techniques

- Signature? MAC? Authenticated encryption? **No**
- Consequence: standard definitions are not applicable.

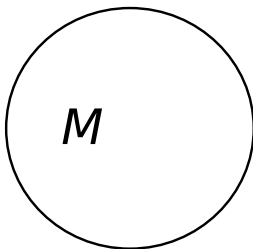
Our Proposal: Robustness Guarantees for Anonymity

- Quantitative definitions for robustness
 - Communication robustness
 - Noise robustness
- Improved Golle-Juels DC-net protocol
 - Improved noise robustness
 - Obviously non-malleable randomized tagging schemes
- Comparison among popular anonymous protocols on robustness
 - Tor, Crowds, Mix Networks, GJ DC-nets

Anonymous communication: scenario

In a single protocol run:

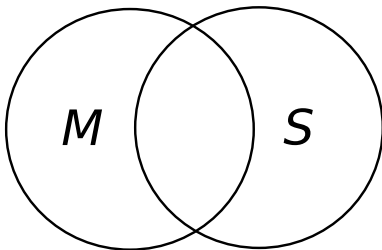
- M is the set (multi-set) of messages supposed to be sent;



Anonymous communication: scenario

In a single protocol run:

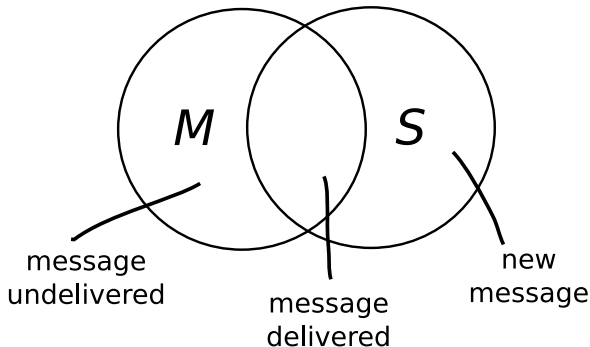
- M is the set (multi-set) of messages supposed to be sent;
- S is the set of messages finally delivered;



Anonymous communication: scenario

In a single protocol run:

- M is the set (multi-set) of messages supposed to be sent;
- S is the set of messages finally delivered;



Communication robustness: motivating example

When participants do not trust each other...

Communication robustness: motivating example

When participants do not trust each other...

Anonymous auction

- Each participant submit an anonymous bid;
- Some participant are malicious and they collude together;
- Highest bid wins the auction.

Communication robustness: motivating example

When participants do not trust each other...

Anonymous auction

- Each participant submit an anonymous bid;
- Some participant are malicious and they collude together;
- Highest bid wins the auction.

Observation

If more honest bids are blocked, then colluded malicious participants have more chance to win the bid with lower price.

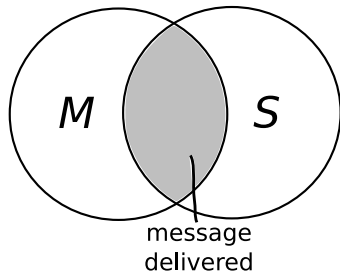
Communication robustness

Communication robustness

Adversary's **inability** to block messages from being delivered.

Observation

- $|S \cap M|$ is the number of message being delivered;
- More robustness if $|S \cap M|$ is bigger.



Noise robustness: motivating example

When participants do not trust each other, again...

Noise robustness: motivating example

When participants do not trust each other, again...

Anonymous Election

- Each member conducts an anonymous vote;
- Colluded malicious members try to block/modify honest members' votes;

Noise robustness: motivating example

When participants do not trust each other, again...

Anonymous Election

- Each member conducts an anonymous vote;
- Colluded malicious members try to block/modify honest members' votes;

Observation

- A successful modification of an honest vote results in:
 - Honest leader losses 1 vote;
 - Malicious leader gains 1 vote;
 - A difference of 2 votes!

Noise robustness

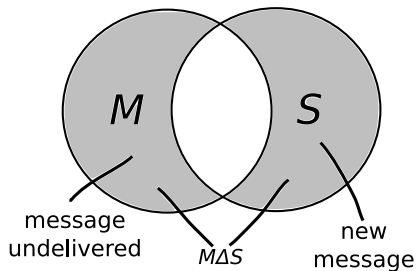
Noise robustness

Adversary's **ability** to interfere messages.

Observation

- $|S\Delta M|$ is the number of message being interfered ^a;
- More robustness if $|S\Delta M|$ is small.

$$^a S\Delta M = (M \setminus S) \cup (S \setminus M)$$



Formal execution model

A protocol (run) include

- Sender parties P_1, P_2, \dots, P_n , a receiver party R ;
- An adversary A ;
- Messages $\vec{M} = M_1, M_2, \dots, M_n$ for each party;
- An execution: $\vec{S} \stackrel{\mathcal{R}}{\leftarrow} \text{Exec}(P(\vec{M}), R, A)(\eta)$
 - η is the security parameter

k -Communication robustness

Intuition

With overwhelming probability, k messages are delivered.

k -Communication robustness

Intuition

With overwhelming probability, k messages are delivered.

Definition (k -Communication Robustness)

$$\Pr \left[\quad \right]$$

k -Communication robustness

Intuition

With overwhelming probability, k messages are delivered.

Definition (k -Communication Robustness)

$$\Pr \left[\vec{M} \stackrel{\mathcal{R}}{\leftarrow} A; \vec{S} \stackrel{\mathcal{R}}{\leftarrow} \text{Exec}(P(\vec{M}), A, R)(\eta) \right]$$

k -Communication robustness

Intuition

With overwhelming probability, k messages are delivered.

Definition (k -Communication Robustness)

$$\Pr \left[|\vec{S} \cap \vec{M}| < k \mid \vec{M} \stackrel{\mathcal{R}}{\leftarrow} A; \vec{S} \stackrel{\mathcal{R}}{\leftarrow} \text{Exec}(P(\vec{M}), A, R)(\eta) \right]$$

k -Communication robustness

Intuition

With overwhelming probability, k messages are delivered.

Definition (k -Communication Robustness)

We define the advantage of an adversary A against the k -communication robustness of protocol P by $\mathbf{Adv}_{P,A}^{k\text{-crob}}(\eta) =$

$$\Pr \left[|\vec{S} \cap \vec{M}| < k \mid \vec{M} \xleftarrow{\mathcal{R}} A; \vec{S} \xleftarrow{\mathcal{R}} \text{Exec}(P(\vec{M}), A, R)(\eta) \right]$$

k -Communication robustness

Intuition

With overwhelming probability, k messages are delivered.

Definition (k -Communication Robustness)

We define the advantage of an adversary A against the k -communication robustness of protocol P by $\mathbf{Adv}_{P,A}^{k\text{-crob}}(\eta) =$

$$\Pr \left[|\vec{S} \cap \vec{M}| < k \mid \vec{M} \stackrel{\mathcal{R}}{\leftarrow} A; \vec{S} \stackrel{\mathcal{R}}{\leftarrow} \text{Exec}(P(\vec{M}), A, R)(\eta) \right]$$

We say that the protocol P is k -communication robust (for a given execution model) if the advantage of any probabilistic polynomial time adversary A is negligible as a function of the security parameter η .

Communication robustness (non-asymptotic def.)

Some protocols may use constant probability during execution (e.g. Tor, Crowds...), where the adversary can always succeed with non-negligible probability.

Definition ((k, ϵ) -Communication Robustness)

We say that protocol P is (k, ϵ) -communication robust if

$$\mathbf{Adv}_{P,A}^{k\text{-crob}} =$$

$$\Pr \left[|\vec{S} \cap \vec{M}| < k \mid \vec{M} \stackrel{\mathcal{R}}{\leftarrow} A; \vec{S} \stackrel{\mathcal{R}}{\leftarrow} \text{Exec}(P(\vec{M}), A, R) \right] \leq \epsilon$$

Noise robustness

Intuition

With overwhelming probability, less than k messages are interfered.

Definition (k -Noise Robustness)

We define the advantage of an adversary A against the k -noise robustness of protocol P by $\mathbf{Adv}_{P,A}^{k\text{-nrob}}(\eta) =$

$$\Pr \left[|\vec{S}\Delta\vec{M}| > k \mid \vec{M} \stackrel{\mathcal{R}}{\leftarrow} A; \vec{S} \stackrel{\mathcal{R}}{\leftarrow} \text{Exec}(P(\vec{M}), A, R)(\eta) \right]$$

We say that protocol P is k -noise robust if the advantage of any probabilistic polynomial time adversary A is negligible as a function of η .

Noise robustness (non-asymptotic def.)

Definition ((k, ϵ) -Noise Robustness)

We say that protocol P is (k, ϵ) -noise robust if $\mathbf{Adv}_{P,A}^{k\text{-nrob}} =$

$$\Pr \left[|\vec{S} \Delta \vec{M}| > k \mid \vec{M} \stackrel{\mathcal{R}}{\leftarrow} A; \vec{S} \stackrel{\mathcal{R}}{\leftarrow} \text{Exec}(P(\vec{M}), A, R) \right] \leq \epsilon$$

Optimal results – communication robustness

Assumptions:

- n participants, t ($t < n$) are corrupted.

Observation

The optimal results for communication robustness is $n - t$.

Why?

- The adversary can always mute corrupted parties.
- The optimal situation is that all honest messages in M are delivered.
- Upper bound of $|S \cap M|$ is $n - t$.

Optimal results – noise robustness

Assumptions:

- n participants, t ($t < n$) are corrupted.

Observation

The optimal results for noise robustness is $2t$.

Why?

- The optimal situation is that the adversary cannot
 - block honest message ($|M \setminus S|$ is at least t)
 - produce more dummy noise ($|S \setminus M|$ is at least t)
- Lower bound of $|S \Delta M|$ is $2t$.

Golle Juels DC-nets protocol

The protocol

- Variant of Chaum's DC-net;
 - Non-interactive generation of keypad by bilinear paring;
 - Use zero-knowledge proof-of-knowledge to ensure correct keypads;
-
- n participants, t corrupted participants;
 - Golle Juels DC-nets is $3t$ -noise robust;
 - Our improvement is $2t$ -noise robust;

Golle-Juels: How does it work?

- Each participant generate a random vector of length n :

 p_1

$$\begin{bmatrix} W_{1,1} \\ W_{1,2} \\ \vdots \\ W_{1,c} \\ \vdots \\ \vdots \\ W_{1,n} \end{bmatrix}$$

$$W_{i,j} = \sum_{k \neq i} \hat{e}(Q_j, y_k)^{\delta_{i,k} \times i}$$

Golle-Juels: How does it work?

- Each $W_{i,j} \in G$ for some G :

$$\begin{array}{cccccc}
 p_1 & p_2 & \dots & p_r & \dots & p_n \\
 \left[\begin{array}{c} W_{1,1} \\ W_{1,2} \\ \vdots \\ W_{1,c} \\ \vdots \\ \vdots \\ W_{1,n} \end{array} \right] & \left[\begin{array}{c} W_{2,1} \\ W_{2,2} \\ \vdots \\ \vdots \\ W_{2,d} \\ \vdots \\ W_{2,n} \end{array} \right] & \dots & \left[\begin{array}{c} W_{r,1} \\ W_{r,2} \\ \vdots \\ W_{r,c} \\ \vdots \\ W_{r,n} \end{array} \right] & \dots & \left[\begin{array}{c} W_{n,1} \\ W_{n,2} \\ \vdots \\ W_{n,c} \\ \vdots \\ W_{n,n} \end{array} \right]
 \end{array}$$

Golle-Juels: How does it work?

- The multiplication of any given row of all vectors is $1 \in G$:

$$\begin{array}{cccccc}
 p_1 & p_2 & \dots & p_r & \dots & p_n \\
 \left[\begin{array}{c} W_{1,1} \\ W_{1,2} \\ \vdots \\ W_{1,c} \\ \vdots \\ \vdots \\ W_{1,n} \end{array} \right] & \left[\begin{array}{c} W_{2,1} \\ W_{2,2} \\ \vdots \\ \vdots \\ W_{2,d} \\ \vdots \\ W_{2,n} \end{array} \right] & \dots & \left[\begin{array}{c} W_{r,1} \\ W_{r,2} \\ \vdots \\ W_{r,c} \\ \vdots \\ W_{r,n} \end{array} \right] & \dots & \left[\begin{array}{c} W_{n,1} \\ W_{n,2} \\ \vdots \\ W_{n,c} \\ \vdots \\ W_{n,n} \end{array} \right] \\
 \Rightarrow & & & & & \left[\begin{array}{c} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{array} \right]
 \end{array}$$

Golle-Juels: How does it work?

- Each participant choose one (possibility of collision) row to multiply (hide) his message:
- To avoid collision: run a slot allocation protocol before.

$$\begin{array}{ccccccc}
 p_1 & & p_2 & & \dots & & p_r & & \dots & & p_n \\
 \left[\begin{array}{c} W_{1,1} \\ W_{1,2} \\ \vdots \\ W_{1,c} \cdot M_1 \\ \vdots \\ \vdots \\ W_{1,n} \end{array} \right] & & \left[\begin{array}{c} W_{2,1} \\ W_{2,2} \\ \vdots \\ \vdots \\ W_{1,d} \cdot M_2 \\ \vdots \\ W_{2,n} \end{array} \right] & & \dots & & \left[\begin{array}{c} W_{r,1} \\ W_{r,2} \\ \vdots \\ W_{r,c} \\ \vdots \\ W_{r,n} \end{array} \right] & & \dots & & \left[\begin{array}{c} W_{n,1} \\ W_{n,2} \\ \vdots \\ W_{n,c} \\ \vdots \\ W_{n,n} \end{array} \right] & \Rightarrow & \left[\begin{array}{c} 1 \\ 1 \\ \vdots \\ M_1 \\ M_2 \\ \vdots \\ 1 \end{array} \right]
 \end{array}$$

Golle-Juels: How does it work?

Zero-knowledge ensures that

- one can only output valid vector;
- one can only attach message in one row;
- **but does not ensure that one follows allocated slot.**

$$\begin{array}{cccccc}
 p_1 & & p_2 & \dots & p_r & \dots & p_n \\
 \left[\begin{array}{c} W_{1,1} \\ W_{1,2} \\ \vdots \\ W_{1,c} \cdot M_1 \\ W_{1,d} \cdot M' \\ \vdots \\ W_{1,n} \end{array} \right] & & \left[\begin{array}{c} W_{2,1} \\ W_{2,2} \\ \vdots \\ \vdots \\ W_{1,d} \cdot M_2 \\ \vdots \\ W_{2,n} \end{array} \right] & & \dots & & \left[\begin{array}{c} W_{r,1} \\ U \\ \vdots \\ W_{r,c} \\ \vdots \\ W_{r,n} \end{array} \right] & & \dots & & \left[\begin{array}{c} W_{n,1} \\ W_{n,2} \\ \vdots \\ \vdots \\ W_{n,c} \\ \vdots \\ \vdots \\ W_{n,n} \end{array} \right]
 \end{array}$$

Analysis: Noise robustness

How can we attack the protocol?

Analysis: Noise robustness

- A corrupted participant wait until all the other participants output their vectors:

$$\begin{array}{ccccccc}
 & p_1 & & p_2 & & \dots & p_r & \dots & & p_n \\
 \left[\begin{array}{c} W_{1,1} \\ W_{1,2} \\ \vdots \\ W_{1,c} \cdot M_1 \\ \vdots \\ \vdots \\ W_{1,n} \end{array} \right] & & \left[\begin{array}{c} W_{2,1} \\ W_{2,2} \\ \vdots \\ \vdots \\ W_{2,d} \cdot M_2 \\ \vdots \\ W_{2,n} \end{array} \right] & & \dots & & \dots & & \left[\begin{array}{c} W_{n,1} \\ W_{n,2} \\ \vdots \\ W_{n,c} \\ \vdots \\ \vdots \\ W_{n,n} \end{array} \right]
 \end{array}$$

Analysis: Noise robustness

- With his own vector, he can compute others' messages:

$$\begin{array}{cccccc}
 p_1 & & p_2 & & \dots & & p_r & & \dots & & p_n \\
 \left[\begin{array}{c} W_{1,1} \\ W_{1,2} \\ \vdots \\ W_{1,c} \cdot M_1 \\ \vdots \\ \vdots \\ W_{1,n} \end{array} \right] & & \left[\begin{array}{c} W_{2,1} \\ W_{2,2} \\ \vdots \\ \vdots \\ W_{2,d} \cdot M_2 \\ \vdots \\ W_{2,n} \end{array} \right] & & \dots & & \left[\begin{array}{c} W_{r,1} \\ W_{r,2} \\ \vdots \\ W_{r,c} \\ \vdots \\ W_{r,n} \end{array} \right] & \left[\begin{array}{c} 1 \\ 1 \\ \vdots \\ M_1 \\ M_2 \\ \vdots \\ 1 \end{array} \right] & & \dots & & \left[\begin{array}{c} W_{n,1} \\ W_{n,2} \\ \vdots \\ W_{n,c} \\ \vdots \\ \vdots \\ W_{n,n} \end{array} \right]
 \end{array}$$

Analysis: Noise robustness

- He can make a collision in purpose, and publishes his vector:

$$\begin{array}{ccccccc}
 p_1 & & p_2 & & \dots & & p_r & & \dots & & p_n \\
 \left[\begin{array}{c} W_{1,1} \\ W_{1,2} \\ \vdots \\ W_{1,c} \cdot M_1 \\ \vdots \\ \vdots \\ W_{1,n} \end{array} \right] & & \left[\begin{array}{c} W_{2,1} \\ W_{2,2} \\ \vdots \\ \vdots \\ W_{2,d} \cdot M_2 \\ \vdots \\ W_{2,n} \end{array} \right] & & \dots & & \left[\begin{array}{c} W_{r,1} \\ W_{r,2} \\ \vdots \\ W_{r,c} \cdot M' \\ \vdots \\ W_{r,n} \end{array} \right] & \left[\begin{array}{c} 1 \\ 1 \\ \vdots \\ M_1 \cdot M' \\ M_2 \\ \vdots \\ 1 \end{array} \right] & & \dots & & \left[\begin{array}{c} W_{n,1} \\ W_{n,2} \\ \vdots \\ W_{n,c} \\ \vdots \\ \vdots \\ W_{n,n} \end{array} \right]
 \end{array}$$

Analysis: Noise robustness

Worst case analysis:

- Given t corrupted participants;
- $|M \setminus S| = 2t$
 - t corrupted messages in \vec{M} are dropped;
 - t honest messages in \vec{M} are block by collision;
- $|S \setminus M| = t$
 - t new messages are produced by collision;

Theorem

Golle-Juels DC-nets is $3t$ -noise robust.

Obliviously non-malleable randomized tagging scheme

Proposed tagging scheme

- A pair of (randomized) algorithm (tag, ver)
- Correctness: if $t \stackrel{\mathcal{R}}{\leftarrow} \text{tag}(m)$ then $\text{ver}(\langle m, t \rangle) = 1$.
- $\langle m, t \rangle$ is reversible encoding of a pair to some group (G, \cdot)

Non-malleability

$\Pr \left[\text{ver}(c \cdot \langle m, t \rangle) = 1 \mid (m, c) \stackrel{\mathcal{R}}{\leftarrow} A; t \stackrel{\mathcal{R}}{\leftarrow} \text{tag}(m) \right]$ is negligible.

Constructions

- $\text{tag}(m) = r \parallel H(m \parallel r)$
- $\text{ver}(\langle m, r \parallel h \rangle)$ check if $h = H(m \parallel r)$

Strong Golle-Juels DC-nets

- Now message are published with padding $t = \text{tag}(M_1)$

$$\begin{array}{c} p_1 \\ \left[\begin{array}{c} W_{1,1} \\ W_{1,2} \\ \vdots \\ W_{1,c} \cdot \langle M_1, t \rangle \\ \vdots \\ \vdots \\ W_{1,n} \end{array} \right] \end{array}$$

Strong Golle-Juels DC-nets

- Simultaneous broadcasting vectors;
- By the property of non-malleability, a collision can only block the message now:

$$\begin{array}{c} p_1 \qquad \dots \qquad p_r \\ \left[\begin{array}{c} W_{1,1} \\ W_{1,2} \\ \vdots \\ W_{1,c} \cdot \langle M_1, t \rangle \\ \vdots \\ \vdots \\ W_{1,n} \end{array} \right] \quad \dots \quad \left[\begin{array}{c} W_{r,1} \\ W_{r,2} \\ \vdots \\ W_{r,c} \cdot C \\ \vdots \\ \vdots \\ W_{r,n} \end{array} \right] \Rightarrow \left[\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \langle M_1, t \rangle \cdot C \neq \langle M', t' \rangle \\ \vdots \\ \vdots \\ \vdots \end{array} \right]
 \end{array}$$

Strong Golle-Juels DC-nets

Worst case analysis:

- Given t corrupted participants;
- t messages in \vec{M} of corrupted participants are dropped ($|M \setminus S|$ is at least t);
- One of the following can happen at most t times:
 - a corrupted participant sends a message not in \vec{M} without collision (increase $|S \setminus M|$ by 1);
 - a corrupted participant produce a collision with an honest participant (increase $|M \setminus S|$ by 1);

Theorem

Strong Golle-Juels DC-nets is $2t$ -noise robust.

Comparison: Results in a nutshell

We apply the definition to existing protocols

	k Communication	τ Noise
Tor	-	-
Tor (comb.)	$(\alpha\mu, \epsilon)$	-
Crowds	-	-
Crowds (comb.)	$(\alpha\mu, \epsilon)$	$((t + 1)n - \alpha\mu, \epsilon)$
Mix Networks ¹	$n - t$	$2t$
GJ DC-Nets	$n - 2t$	$3t$
SGJ DC-Nets	$n - 2t$	$2t$

¹For mix networks in [PIK93] and [JJ01]

Thank you!

Q & A time.

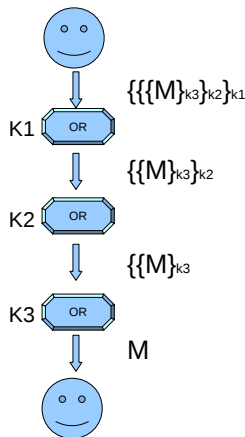
Backup slides

Tor

Tor network implements The Onion Routing protocol.

Protocol description

- 1 m Onion Routers: OR_1, OR_2, \dots, OR_m ;
- 2 Alice build a circuit $OR_{i_1} \dots OR_{i_n}$ for sending messages, each router only knows its predecessor and successor.
- 3 Alice shares a symmetric key K_{i_j} with each router along the circuit in the circuit.
- 4 Operates as an onion.



Tor analysis (1)

Assumptions:

- $0 < p < 1$ fraction of routers are corrupted. Each path contain ℓ routers.

How can adversary not to block honest messages?

By not having at even one corrupted router along the path.

- The probability is $(1 - p)^\ell$

Results

- For asymptotic definition, the probability is constant;
- For combinatorial definition, chernoff bound can give good approximation;

Tor analysis (2)

Assumptions:

- $0 < p < 1$ fraction of routers are corrupted. R accepts unlimited number messages.

How can adversary modify honest messages or generate dummy messages?

By having at least one corrupted router along the path.

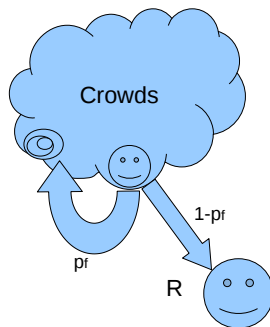
Or by generating dummy messages.

- For either asymptotic definition or combinatorial definition, dishonest router can always fill R with unlimited number of messages, thus any noise robustness will not hold.

Crowds

Protocol description

- 1 n users (called jondos) in the crowd, t are corrupted.
- 2 A honest jondo behaves as follows:
 - With probability p_f , it forwards received message to a random jondo in the crowds ;
 - With probability $1 - p_f$, it forwards the message to R .



Crowds analysis (1)

Assumptions:

- Each jondo initiate one message. t among n jondos are corrupted.

How can not adversary block honest messages?

By not having at even one corrupted jondos along the path.

- The probability is $\frac{n(1-p_f)}{n-p_f(n-t)}$

Results

- For asymptotic definition, the probability is constant;
- For combinatorial definition, chernoff bound can give good approximation;

Crowds analysis (2)

Assumptions:

- R receives **at most** n messages from each jondo.

How can adversary modify honest message or generate noise?

By having having at least one corrupted router along the path.
Or by sending dummy messages.

- Note that R limits the power of dummy messages;

Analysis for combinatorial noise robustness

- $S \Delta M = S \setminus M + M \setminus S$
- $S \setminus M$ is bounded by R
- $M \setminus S$ is bounded by R

Mix networks

Protocol decryption

- 1 Trusted mix-net server with public key PK
- 2 n users encrypt their message with PK , and send them to mix-net server
- 3 When mix-net server collect all n messages it decrypts them and send them to R (the order of message is shuffled).

Robustness results

$n - t$ for comm. robustness, $2t$ for noise robustness.

- Get optimal results with **strong** assumptions.

Golles Juels DC-nets

We also improved Golles Juels DC-nets protocol for better robustness results:

Golles Juels DC-nets

- $n - 2t$ for communication robustness
- $3t$ for noise robustness

Strong Golles Juels DC-nets

- $n - 2t$ for communication robustness
- $2t$ for noise robustness
- By using non-malleable randomized tagging schemes