Selective Diffusion of Ratings in Trust Propagation for MANETs

Dang Q. Nguyen, Louise Lamont and Peter C. Mason

Communications Research Centre Canada
An Agency of Industry Canada

Centre de recherches sur les communications Canada
Un organisme d'Industrie Canada

Outline

Outline

What is a MANET?

Mobile *ad hoc* networks

- Infrastructureless $\longrightarrow$ easy to deploy, fault tolerant.
- Wireless communications with limited bandwidth, average user mobility.
- Distributed processing.
- Multihop routing protocols (e.g. OLSR) to route information from one end to another end of MANET via intermediate users acting as relays.

Trust propagation

Trust relationship in MANET

The actions of each MANET user are observed and rated by its direct neighbours. This rating (trust metric) of a user can be defined as the *consistency* of a user's behaviour.
Trust in MANET environment:

- Distributed processing $\rightarrow$ one user can have different ratings according to different neighbours $\rightarrow$ how to combine those ratings?
- Infrastructureless, user mobility $\rightarrow$ how to share (propagate) the ratings with remote users?

Challenges

- Self-correcting behaviour.
- Colluding cheaters.

Outline

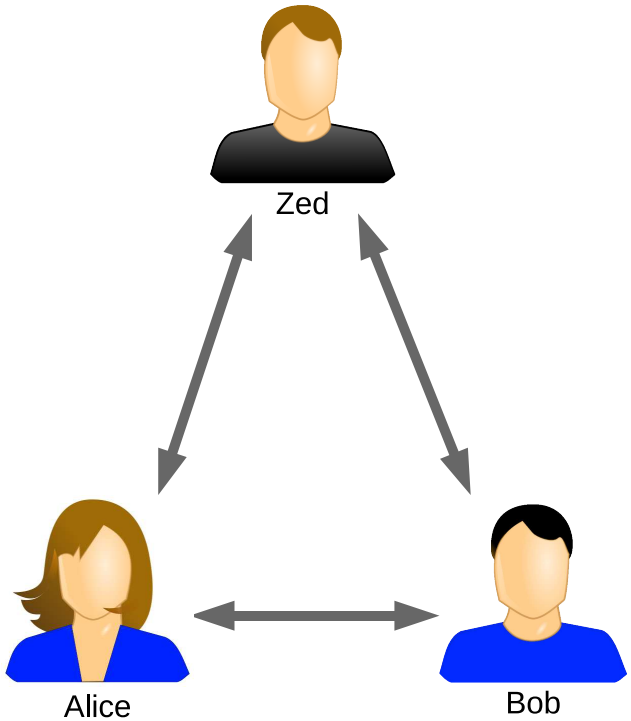
Self-correcting behaviour

Problem

- Alice, Bob and Zed interact with each other.
- They observe the results of their interactions and note their mutual trust ratings.
- They broadcast their notations.
- Zed reads its trust ratings and adapt its behaviour to appear less threatening in the future.

Solution

Alice and Bob use encryption keys to communicate.



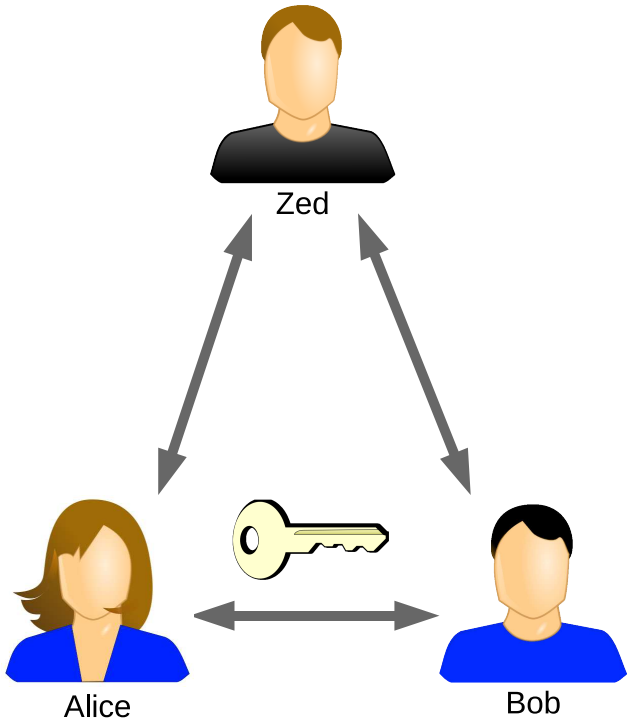
Self-correcting behaviour

Problem

- Alice, Bob and Zed interact with each other.
- They observe the results of their interactions and note their mutual trust ratings.
- They broadcast their notations.
- Zed reads its trust ratings and adapt its behaviour to appear less threatening in the future.

Solution

Alice and Bob use encryption keys to communicate.



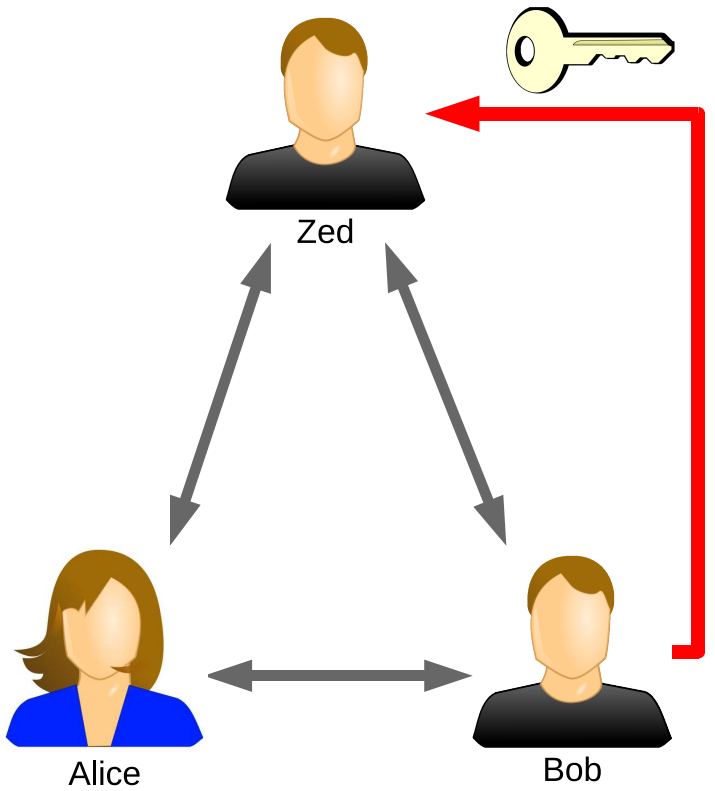
Colluding cheaters

New problem

- Bob turns malicious and joins Zed.
- Bob gives Alice's key to Zed.
- Zed can now decrypt all what Alice says.

Solution

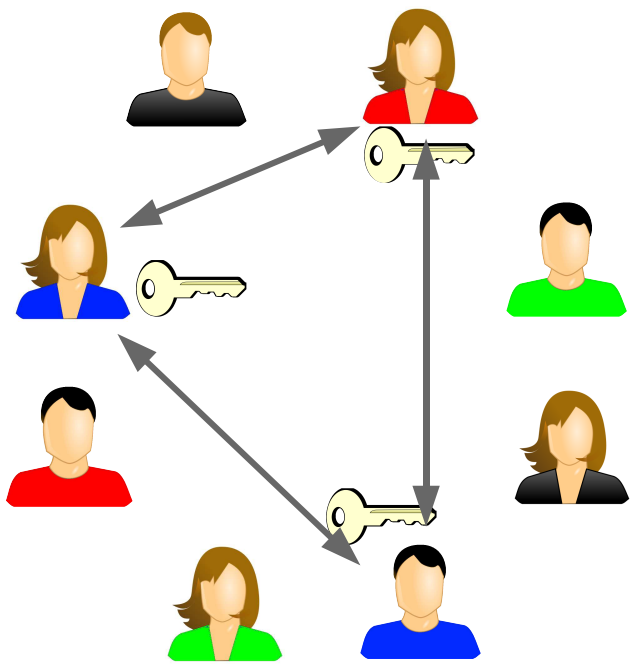
Deter Bob from sharing keys.



Group keys

Group communications

- Group keys are used in subgroup's communications of a group.
- There are 2^n distinct subgroups in a group of n members.
- Boyd generalized RSA cryptosystem to n users: only n keys are needed to encrypt/decrypt communications for 2^n subgroups.
- Every member in a subgroup shares the same subgroup key.



New problem

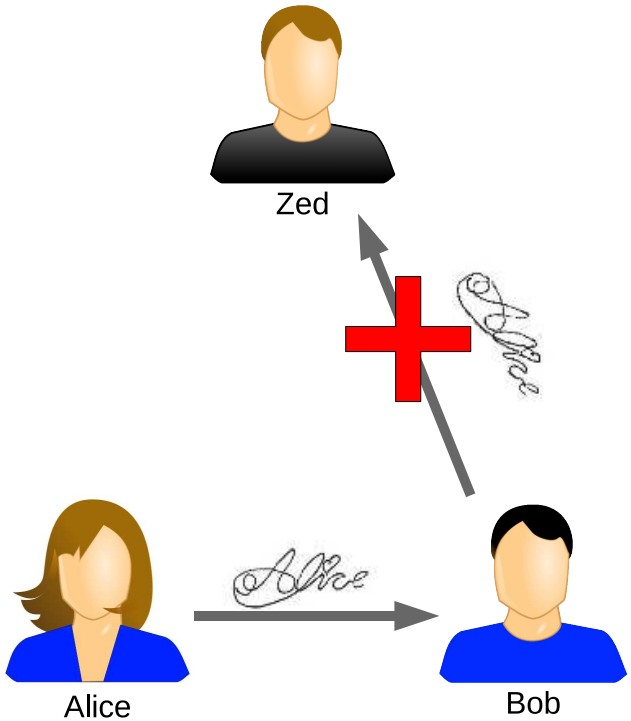
What if a member reveals its subgroup keys?

Undisclosable information

Dedicated digital signature (*dds*)

Klonowski et al. (2006):

- Alice sends a message (M) to Bob, along with her *dds* created from M.
- Bob uses his secret key to obtain the Alice's standard signature for M from the *dds*.
- If Bod shows this standard signature to Zed then Zed can calculate Bob's secret key, if Zed also knows Alice's *dds*.



Outline

Keys scheme

Group keys for selective broadcast

n users, $n+1$ keys, each user keeps n keys:

- k_i is the symmetric key that is used to encrypt/decrypt all ratings of i^{th} user.
- k_i is known to all users, except i^{th} user.
- k^* is the symmetric key used to encrypt/decrypt the broadcasted messages for all n users: k^* is known to all n users.

Each user keeps n symmetric keys: $k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_n$ and k^*.

Keys distribution protocol

When a new user joins the group

The key server calculates new keys as follows.

1. The new user –user $(n+1)^{th}$– is authenticated by the server.
2. The server generates a new broadcast key k'^* and sends it to $(n+1)^{th}$ along with all the n symmetric keys: $k_1 \oplus k^*, \ldots, k_n \oplus k^*$.
3. The server generates a new symmetric key $k_{(n+1)}$, used to encrypt/decrypt $(n+1)^{th}$'s ratings, and broadcast it with k'^* to all n existing users in the group. This message is encrypted using k^*.
4. For all n existing users: k_i is replaced by $k_i \oplus k^*$ and k^* is then replaced by k'^*.

Backward security

Backward security is preserved: $(n+1)^{th}$ user cannot read old messages.

Keys distribution protocol

When an existing user leaves the group

n^{th} user is leaving the group.

1. The server generates a random parameter r and sends it to $n-1$ other users, encrypted with k_n.
2. For all $n-1$ users: k_i is replaced by $k_i \oplus r$ and k^* is then replaced by $k^* \oplus r$.

Forward security

Forward security is preserved: n^{th} user cannot read future messages.

Undisclosable keys and collusion prevention

Construction of undisclosable keys

based on ElGamal's encryption scheme (1985). Let $x_b, y_b = \sigma^{x_b} \mod p$ be the secret and public keys of user Bob. The server computes:

$$\begin{cases} \phi &= \sigma^k \mod p \\ \omega &= y_z^k x_b \mod p. \end{cases}$$

$\hat{k}_b(z) = (\phi, \omega)$ will be the first of the two keys used by Bob to decrypt all ratings of Zed.

Collusion prevention

If Bob shows $\hat{k}_b(z)$ to Zed then Zed can compute Bob's secret key: $x_b = \omega / \phi^{x_z} \mod p$.

Encryption/Decryption of ratings

Keys

For every pair of users (Alice,Zed), the server chooses two large numbers at random $M_a(z), c_a(z)$ such that:

$$\forall i = 1, \ldots, n \text{ and } i \neq a, z : M_a(z) > c_a(z)\hat{k}_i(z).$$

The server then computes:

$$\forall i = 1, \ldots, n \text{ and } i \neq a, z : \hat{r}_{(a,i)}(z) = M_a(z) - c_a(z)\hat{k}_i(z).$$

$\hat{r}_{(a,i)}(z)$ will be the second of the two keys used by i^{th} user to decrypt all ratings of Zed that are rated by Alice. The first key was $\hat{k}_i(z)$.

Encryption/Decryption of ratings

Encryption

- Alice generates a one-time encryption key $k(z) = \alpha^{M_a(z)+r} \mod p$ to encrypt Zed's ratings.
- The ciphertext is broadcasted along with $(\alpha, p, \beta, \gamma)$, with $\beta = \alpha^{c_a(z)} \mod p$ and $\gamma = \alpha^r \mod p$.

Encryption/Decryption of ratings

Decryption

Bob, or any other user, recovers the key $k(z)$ with the following calculations:

$$\begin{aligned} k(z) &= \left(\alpha^{\hat{r}_{(a,b)}(z)} \mod p\right) . \left(\beta^{\hat{k}_b(z)} \mod p\right) .\gamma \mod p \\ &= \alpha^{\hat{r}_{(a,b)}(z)+c_a(z)\hat{k}_b(z)+r} \mod p \\ &= \alpha^{M_a(z)+r} \mod p. \end{aligned}$$

Bob can then use $k(z)$ to decrypt Zed's ratings.

Summary of encryption/decryption

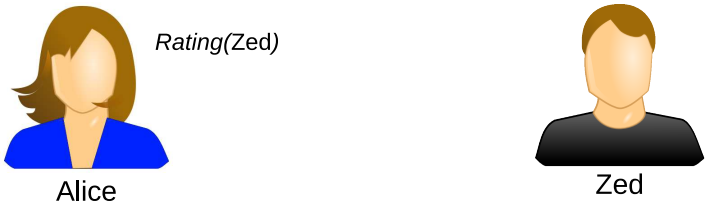


Figure: Alice observes and rates Zed's behaviour.

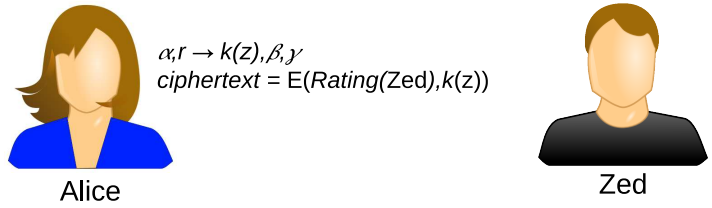
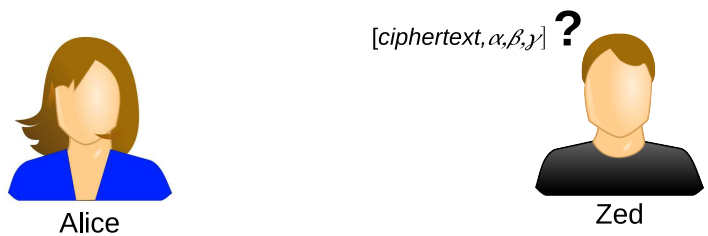


Figure: Alice chooses α, r at random, then computes $k(z), \beta, \gamma$ and encrypts Zed's ratings with $k(z)$.



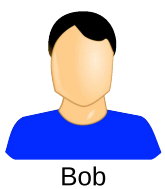


Figure: Alice broadcasts the ciphertext along with α, β, γ.

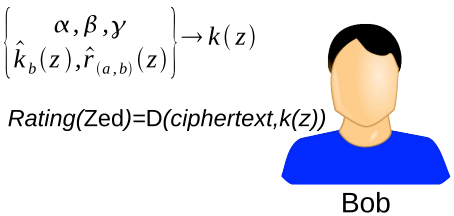


Figure: Bob can compute $k(z)$ using its keys $\hat{k}_b(z), \hat{r}_{(a,b)}(z)$. Zed cannot compute $k(z)$.

Outline

Mitigating attacks

Trust-based models

Blackmail attack: a malicious user threatens to ruin the reputation of another user should that user report negatively upon it.

- Here, malicious user is unable to read reports on its behaviour so it is unaware of what being reported.

Self-correcting behaviour:

- A malicious user cannot adapt its behaviour according to its ratings because it cannot read its ratings.

Colluding cheaters:

- Colluding cheaters can share secrets about each other, but not the keys to systematically decrypt those secrets.

Outline

Conclusion

New encryption scheme for MANET

- performs selective diffusion of information,
- prevents colluding malicious users,
- can be used to enhance trust-based models in MANET.