

Introducing Man In The Contacts attack to trick encrypted messaging apps



Cyber Security Conference #CYBSEC16

03/11/2016 – Securing Apps

whois securिंगapps

- Developer background
- Spent last 10 years working in Switzerland on security products and solutions
 - Focus on mobile since 2010
- Now software security consultant at my own company
 - <http://www.securिंगapps.com>
- Provide services to build security in software
 - Mobile
 - Web
 - Cloud
 - Internet Of Things
 - Bitcoin/Blockchain



@SecuringApps



Introduction

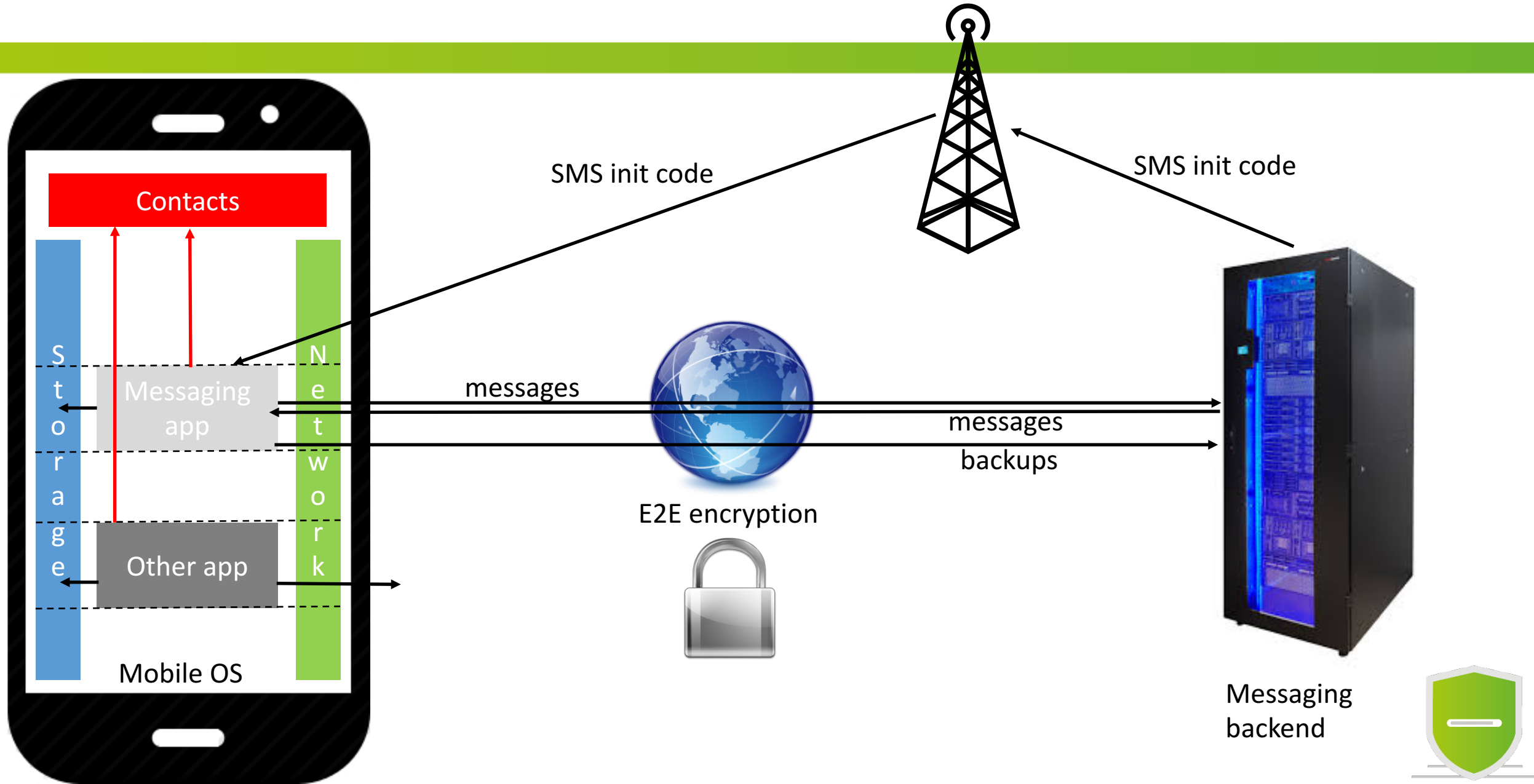
- Popular messaging apps recently switched to End-to-End encryption
 - Great communication around it
 - Privacy now is a requirement
- Debates at the government level to ask for backdoors
 - Going dark ?
 - Used by terrorists ?
- Increased feeling that those applications are unbreakable



- _____



Threat model: mobile focus & simplified



Accessing contacts

- Easy to read/modify/create contacts

- There is an API for that
- Android example

```
private boolean updateContactName(String phone, String newName) {
    ArrayList<ContentProviderOperation> ops = new ArrayList<ContentProviderOperation>();

    ops.add(ContentProviderOperation.newUpdate(ContactsContract.Data.CONTENT_URI)
        .withSelection(ContactsContract.CommonDataKinds.Phone.NUMBER + "=?", new String[]{String.valueOf(phone)})
        .withValue(ContactsContract.CommonDataKinds.StructuredName.DISPLAY_NAME, newName)
        .build());

    try {
        getContentResolver().applyBatch(ContactsContract.AUTHORITY, ops);
        return true;
    } catch (Exception e) {
        Log.e("oups", "aie", e);
    }
    return false;
}
```

- Shared data structure accessible in read/write

- Only restricted by permissions
- And it contains authentication data in clear !

```
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
```

- There is **room for a side channel attack: Man In The Contacts**

- Not requiring a rooted device (e.g no RowHammer attack)



Introducing Alice, Bob and Eve

- Convention: Alice on the left, Bob on the right, Eve in the center
- **Devices not rooted**, latest OS updates available
- Installed apps: latest version (29th October 2016) of **WhatsApp**, **Telegram** and **Signal**



Alice
+33 X XX XX XX 60



Eve
+41 XX XXX XX 21

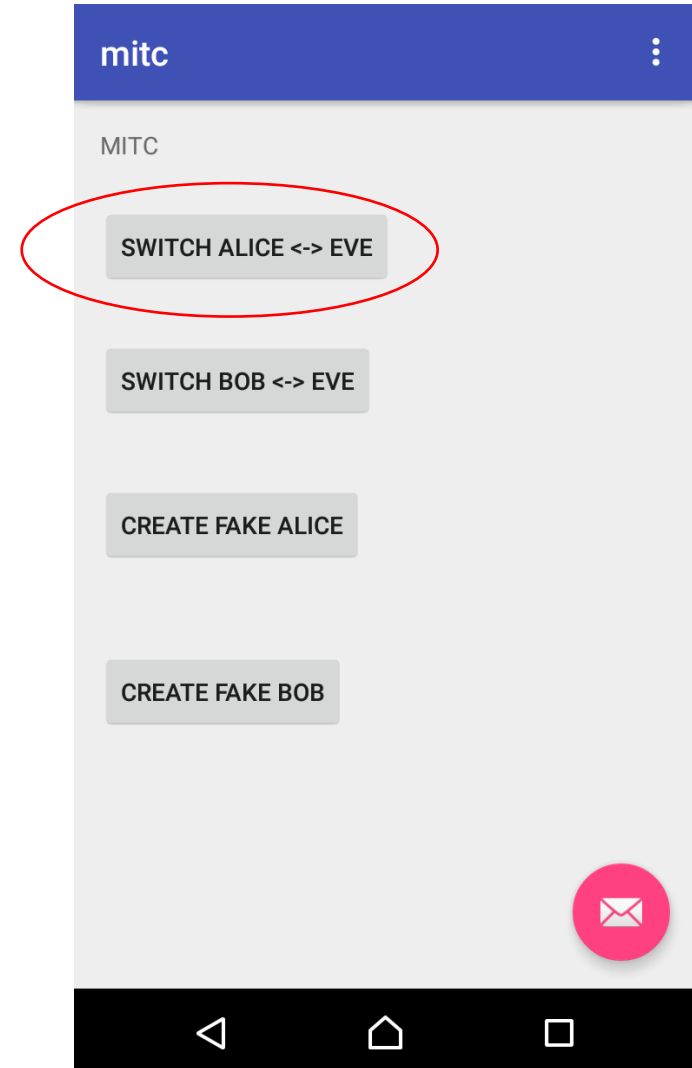


Bob
+41 XX XXX XX 66



Old joke: swap contacts

- Install MITC app on Bob's device
- Start a conversation between Alice and Bob
- Swap Alice and Eve phone numbers on Bob's device
- See what happens



Bob



Old joke: swap contacts

- WhatsApp 1
- Bob starts a conversation with Alice
- Alice answers

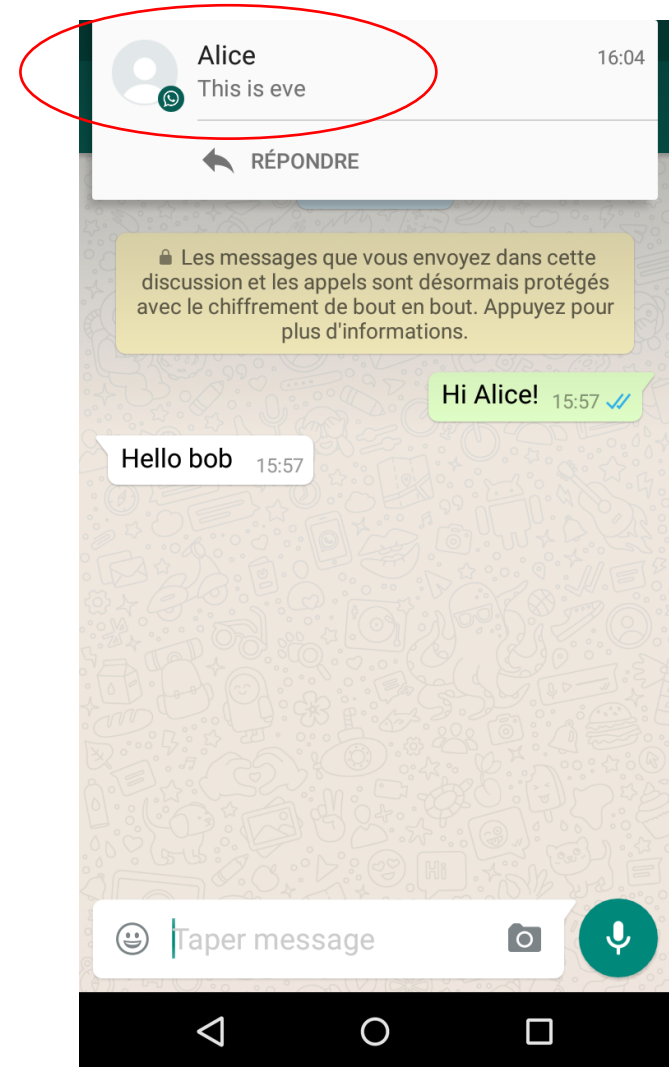


Bob



Old joke: swap contacts

- WhatsApp 2
- Eve triggers contact swap via remote MITC app on Bob's device
- Eve sends «This is eve» to Bob
- Notification received as Alice
- But new conversation

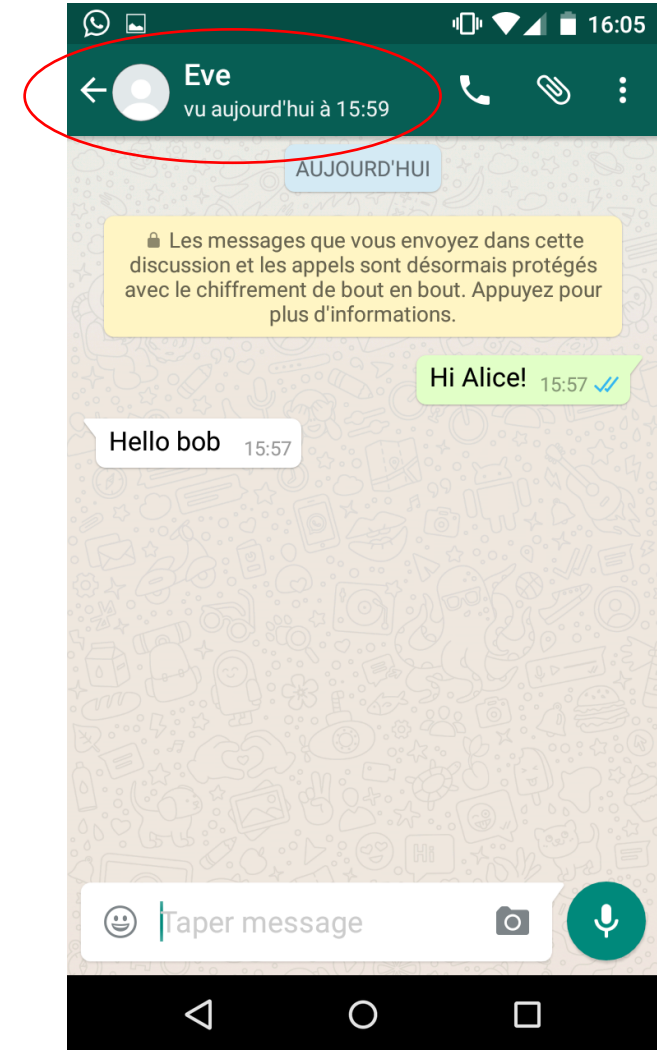


Bob



Old joke: swap contacts

- WhatsApp 3
- Ignore notification
- Conversation of Alice now displayed as Eve

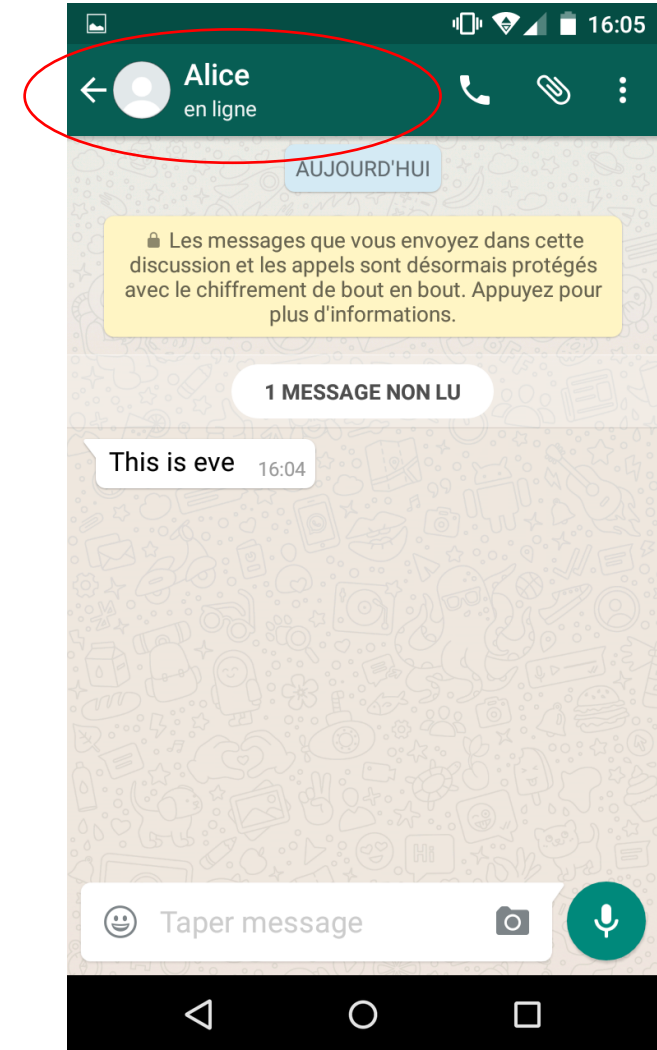


Bob



Old joke: swap contacts

- WhatsApp 4
- Accept notification
- Eve triggered a new conversation
- But displayed as Alice

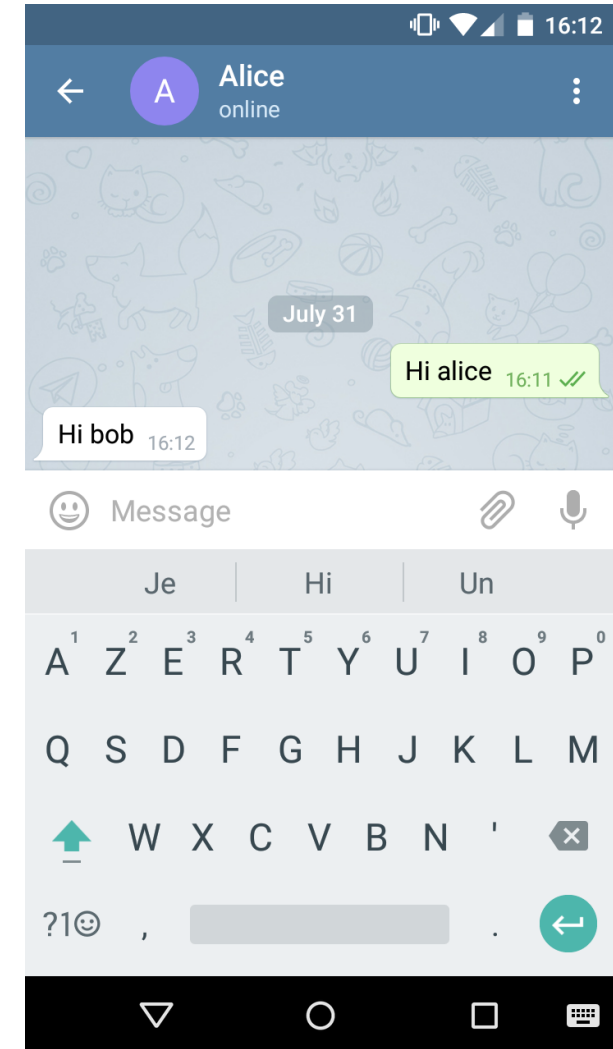


Bob



Old joke: swap contacts

- Telegram 1
- Bob starts a conversation with Alice
- Alice answers

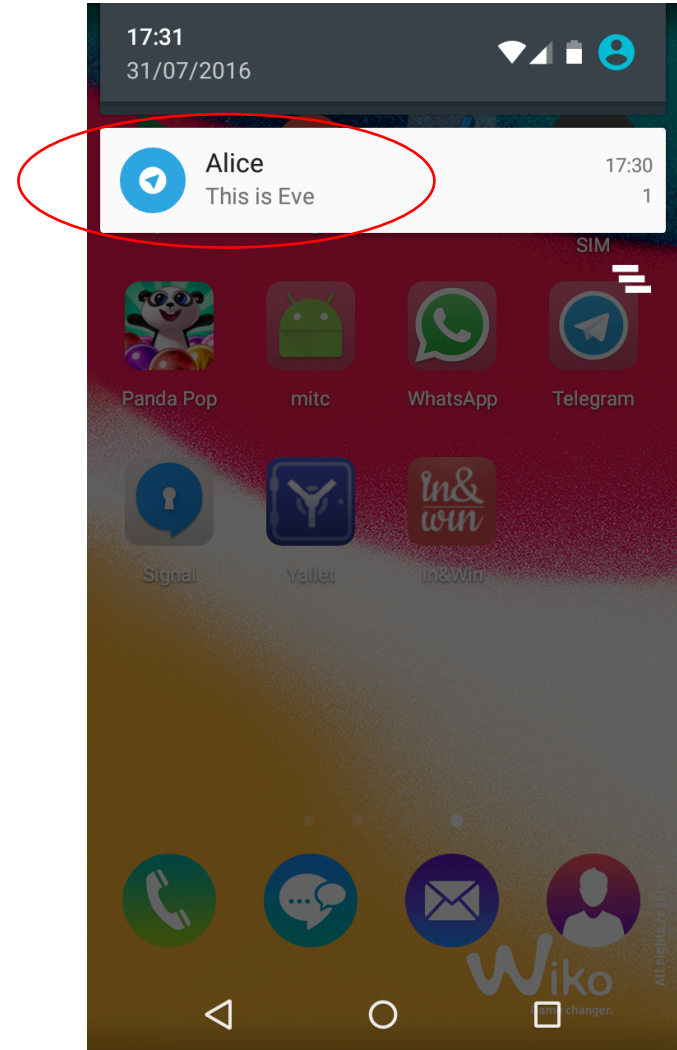


Bob



Old joke: swap contacts

- Telegram 2
- Eve triggers contact swap via remote MITC app on Bob's device
- Eve sends «This is Eve» to Bob
- Notification received as Alice
- But new conversation

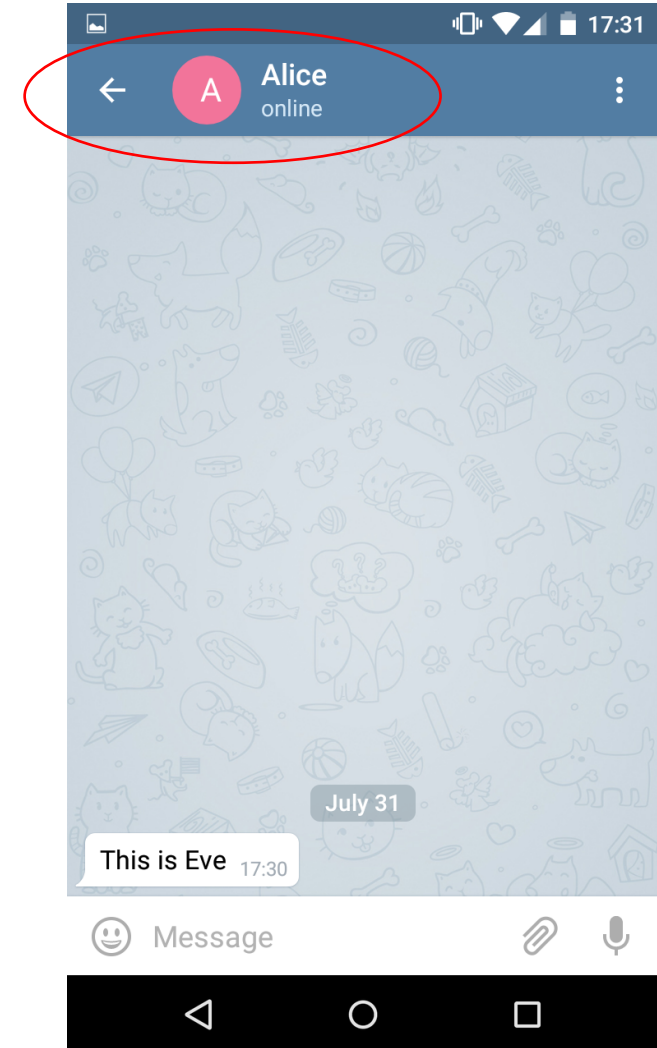


Bob



Old joke: swap contacts

- Telegram 3
- Accept notification
- Eve triggered a new conversation
- But displayed as Alice
- NB: If you change the name of Alice, in the future notifications and conversations will still be under the name of Alice

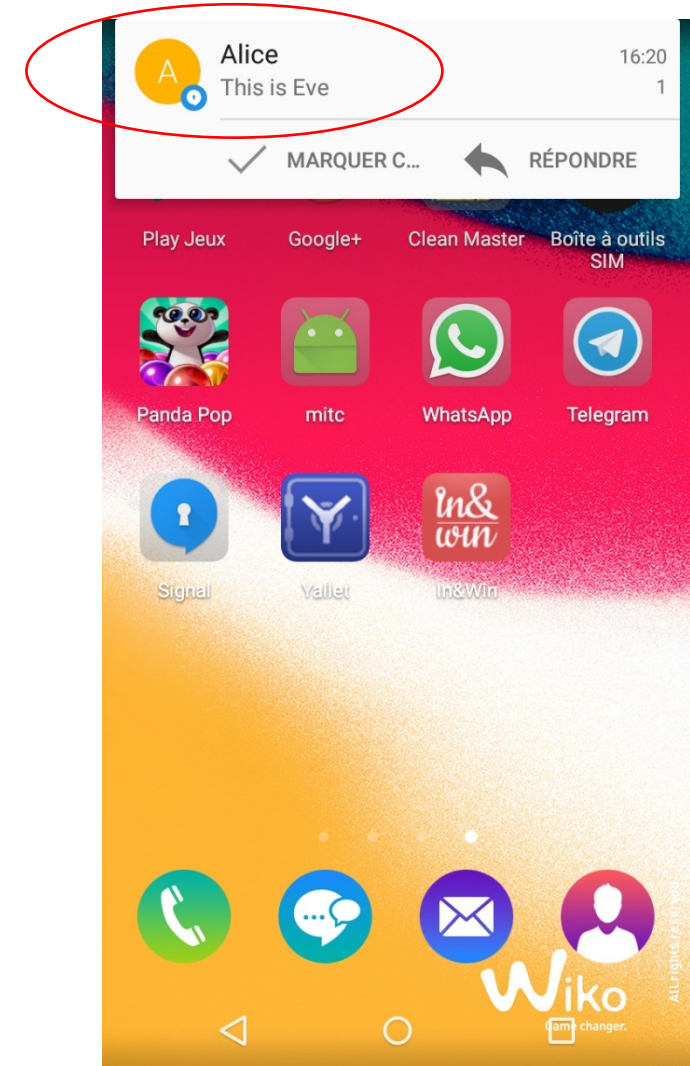


Bob



Old joke: swap contacts

- Signal 1 & 2
- Screenshots refused by Android app
- But same behaviour than WhatsApp

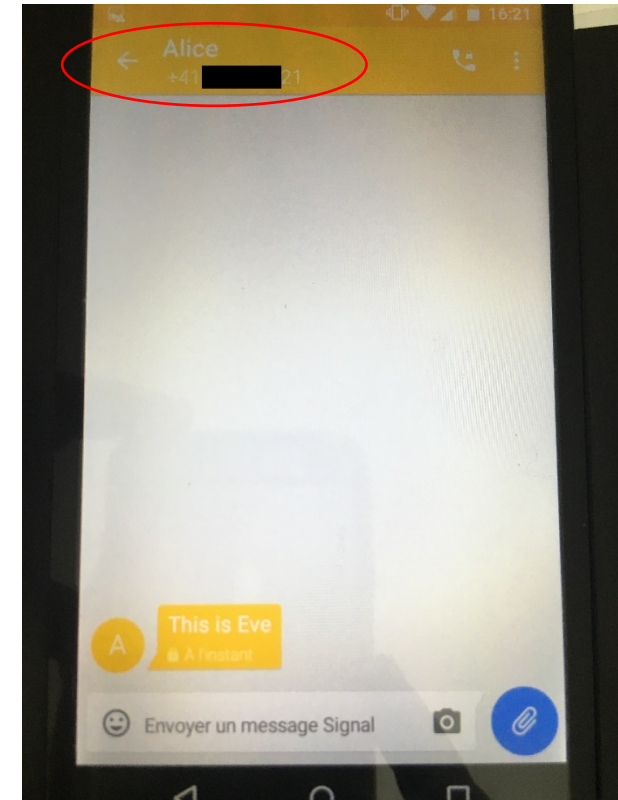


Bob



Old joke: swap contacts

- Signal 3
- Accept notification
- Eve triggered a new conversation
- Displayed as Alice
- But phone number also displayed (the iOS version doesn't display it)

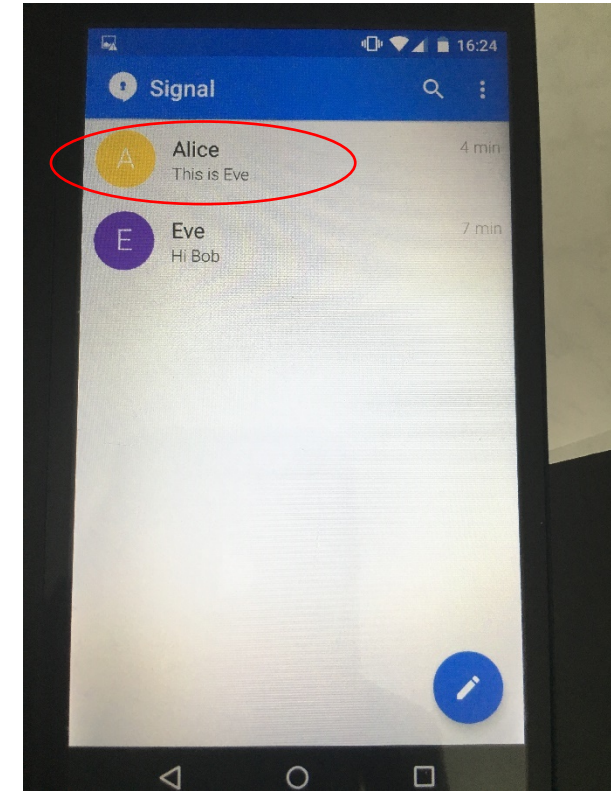


Bob



Old joke: swap contacts

- Signal 4
- Not the case in the main view

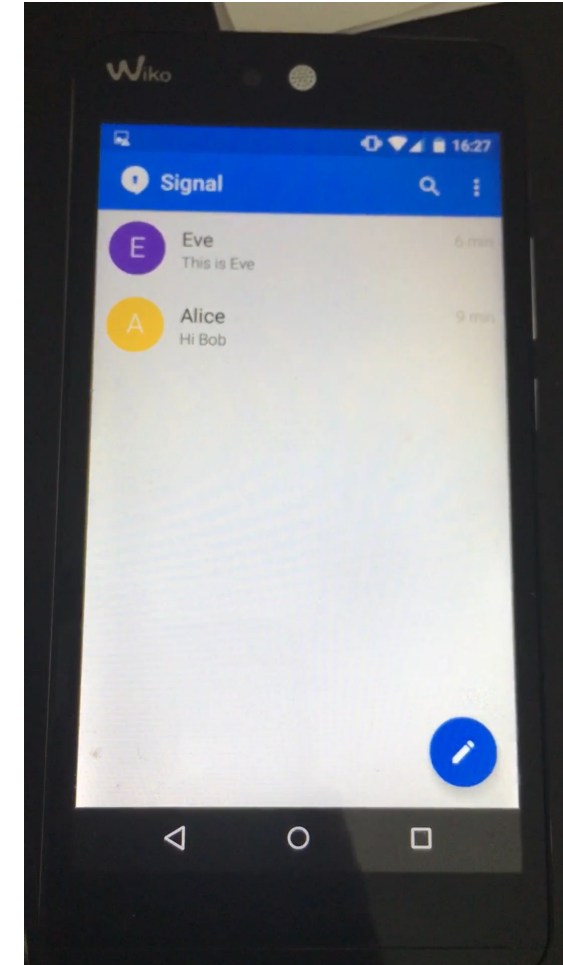


Bob



Old joke: swap contacts

- Signal 5
- Stay in this view
- Switch back contacts with MITC app
- Nothing happens for a while
- And then main view updated
=> contact sync process



Bob



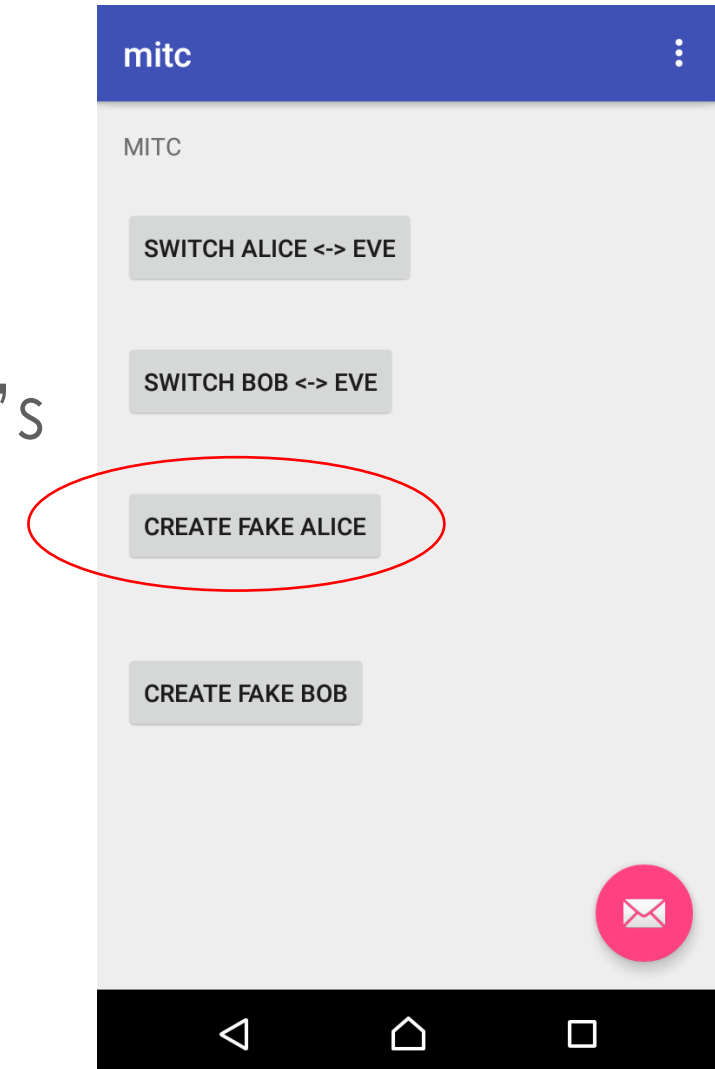
Swap contacts results

- Can't be used to trick Bob within an existing conversation
- But produces a notification and a new conversation that may seem legitimate to Bob
- Different behaviors depending on the app
 - Name in the notification vs name in the conversation
 - Name configured by the sender vs contact name as seen by receiver
 - Contact sync time
- Not discrete in case Alice and Bob have a phone call or send a message



Nasty trick: contact with similar name

- Start a conversation between Alice and Bob
- Create a contact name « **Alice** » on Bob's device with Eve's phone number
- See how the **whitespace** in front of Alice gets displayed

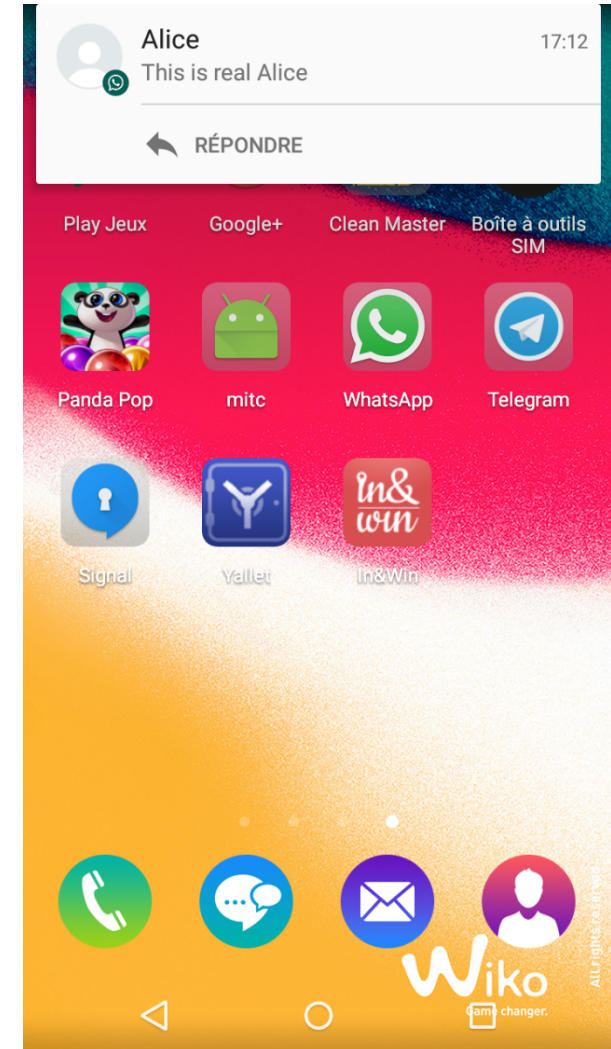


Bob



Nasty trick: contact with similar name

- WhatsApp 1
- Alice starts a conversation with Bob



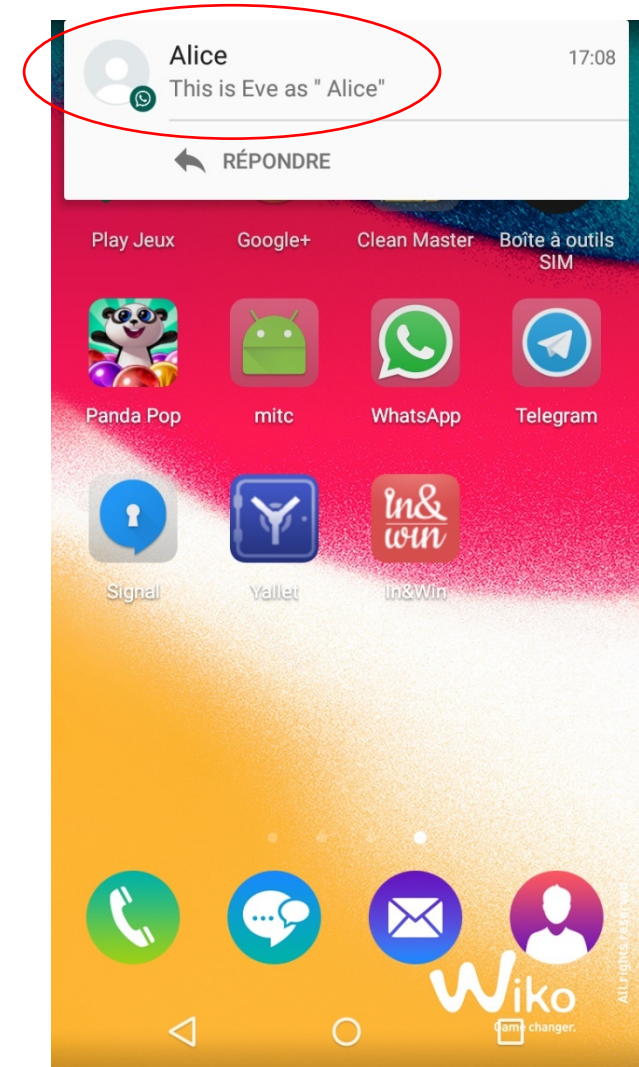
Bob



Nasty trick: contact with similar name

- WhatsApp 2

- Eve starts a conversation with Bob



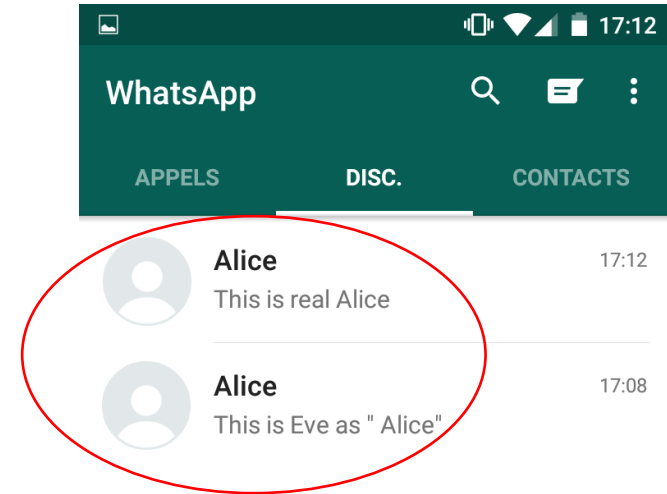
Bob



Nasty trick: contact with similar name

- WhatsApp 3

- Main view



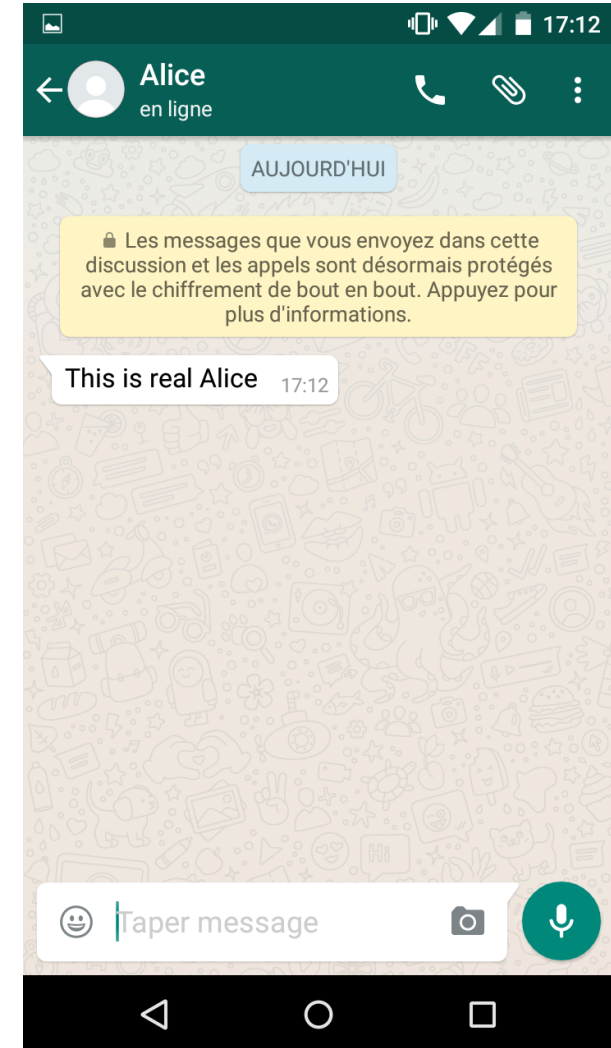
Bob



Nasty trick: contact with similar name

- WhatsApp 4

- Conversation with real Alice



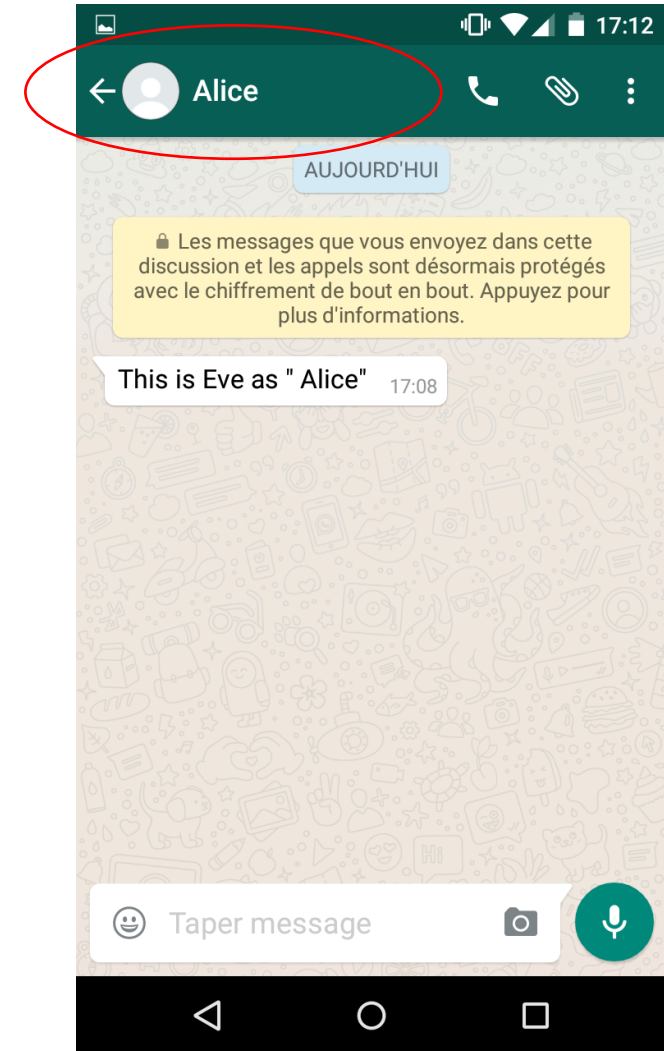
Bob



Nasty trick: contact with similar name

- WhatsApp 5

- Conversation with Eve as « Alice »

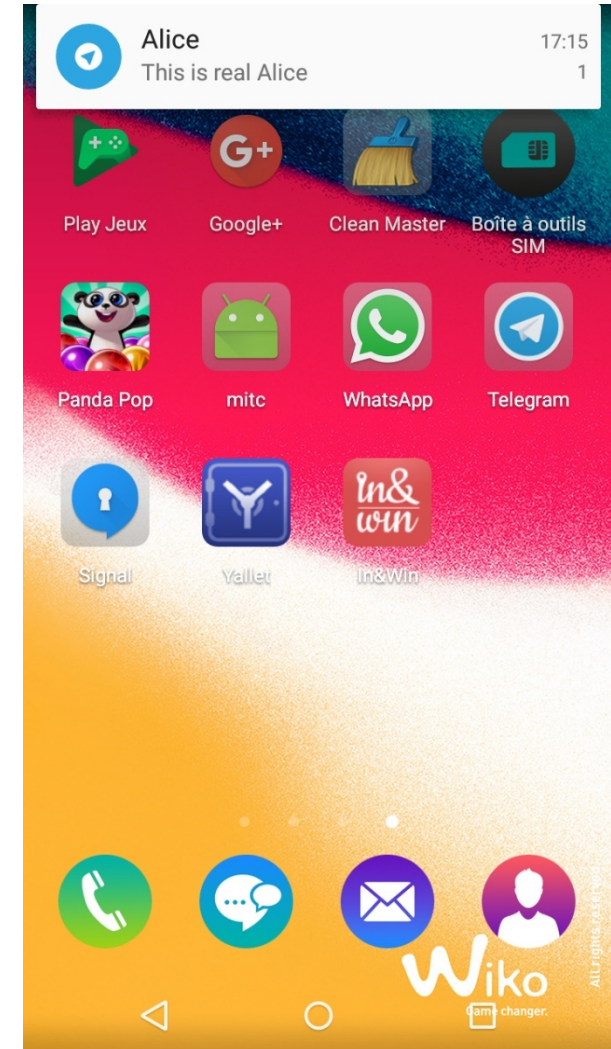


Bob



Nasty trick: contact with similar name

- Telegram 1
- Alice starts a conversation with Bob

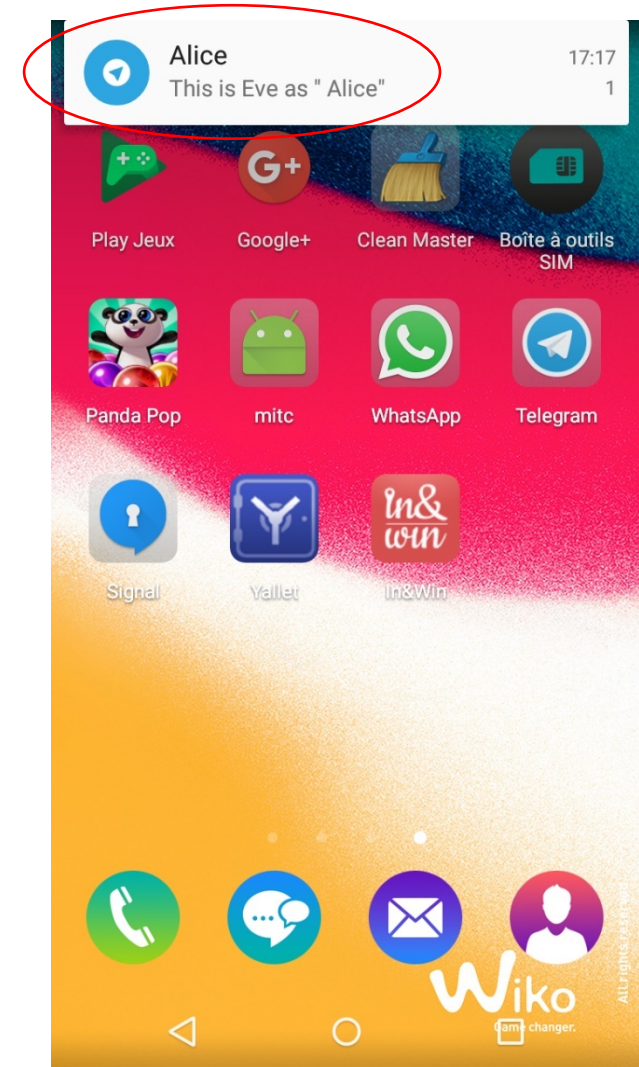


Bob



Nasty trick: contact with similar name

- Telegram 2
- Eve starts a conversation with Bob



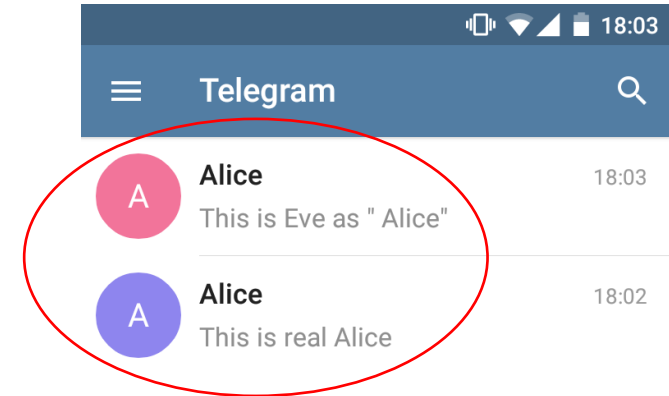
Bob



Nasty trick: contact with similar name

● Telegram 3

● Main view

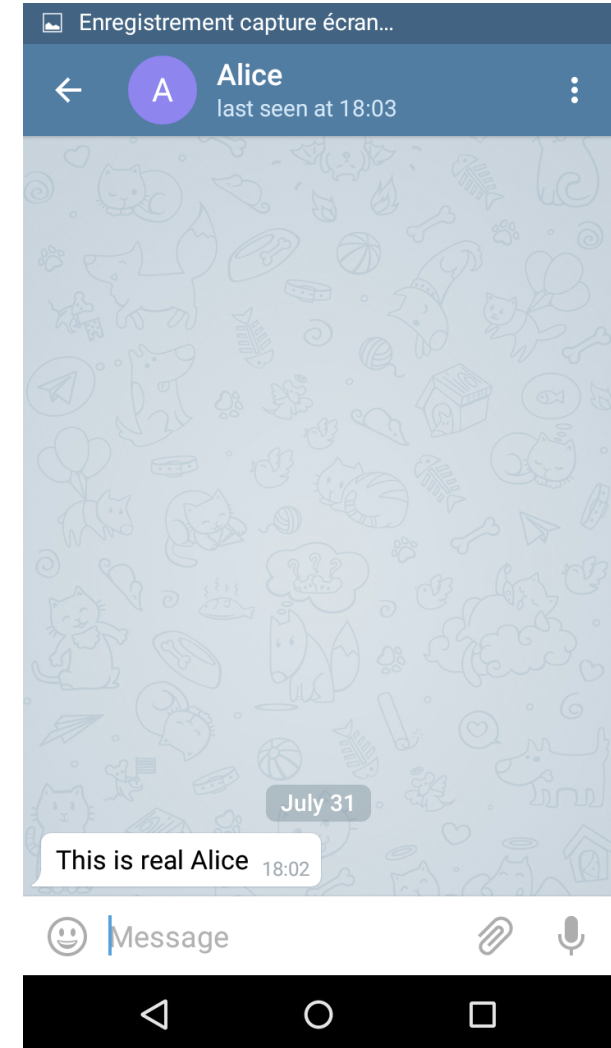


Bob



Nasty trick: contact with similar name

- Telegram 4
- Conversation with real Alice

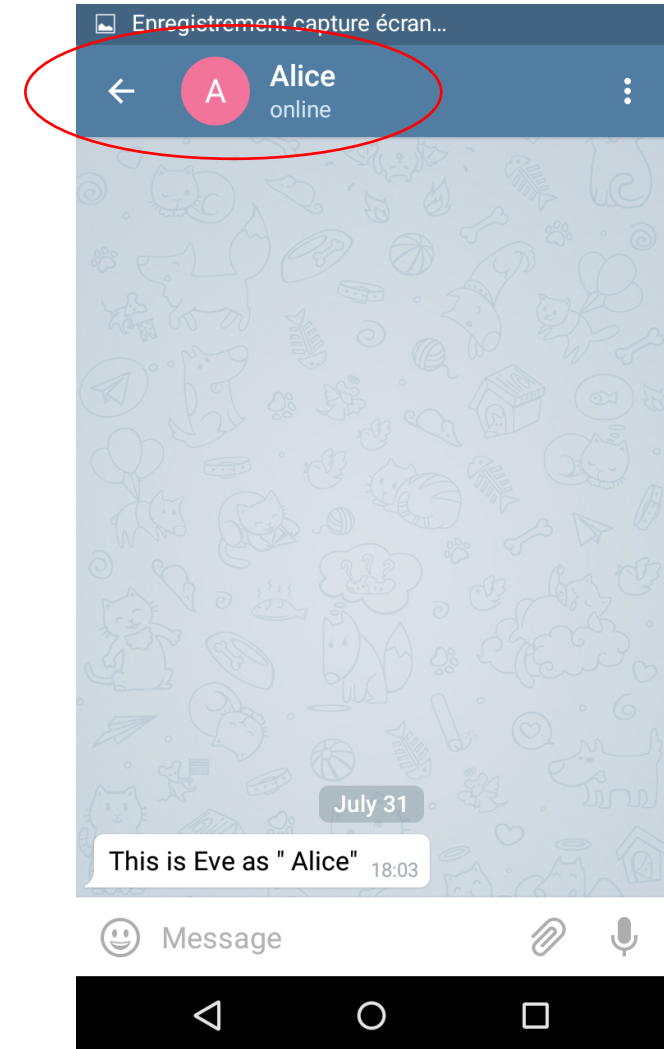


Bob



Nasty trick: contact with similar name

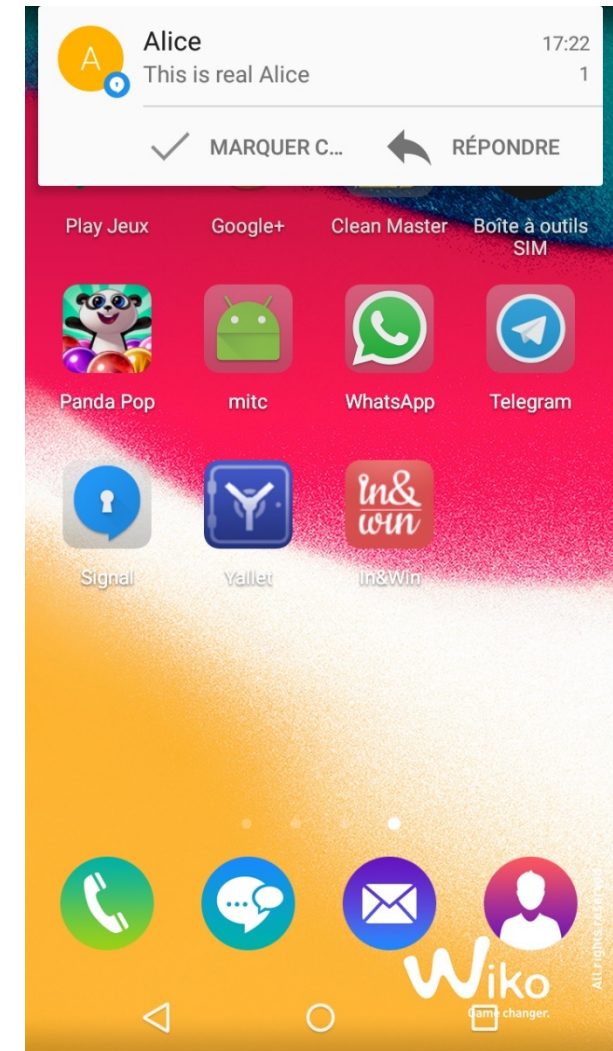
- Telegram 5
- Conversation with Eve as « Alice »



Bob

Nasty trick: contact with similar name

- Signal 1
- Alice starts a conversation with Bob



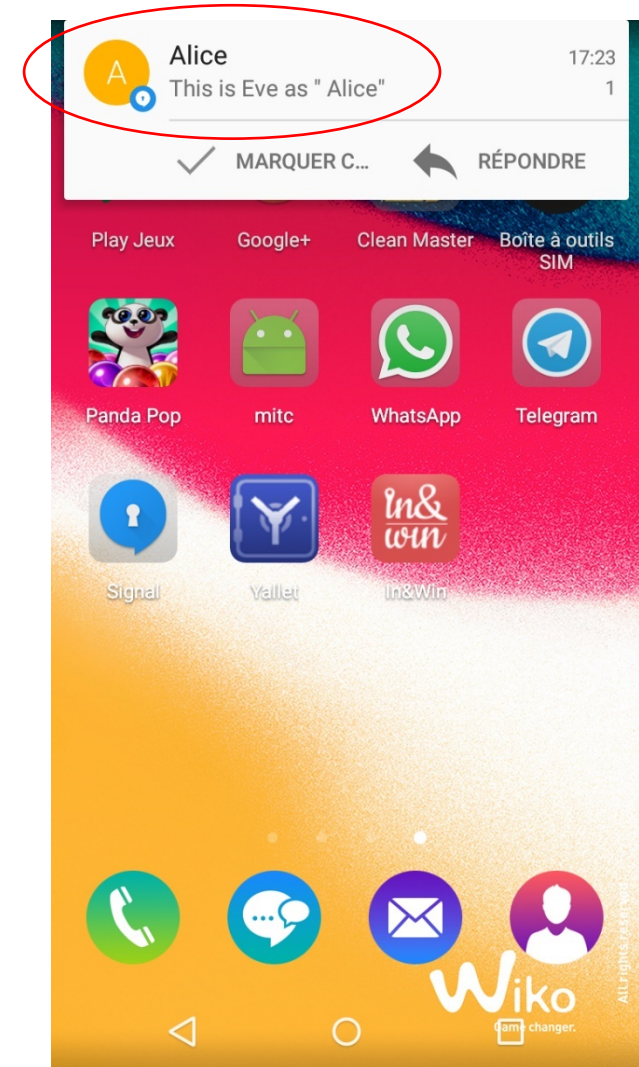
Bob



Nasty trick: contact with similar name

- Signal 2

- Eve starts a conversation with Bob

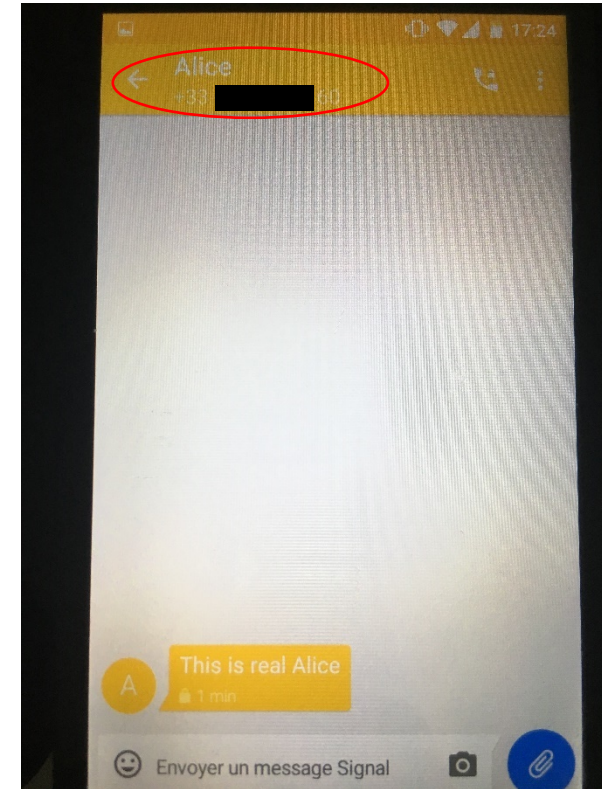


Bob



Nasty trick: contact with similar name

- Signal 4
- Conversation with real Alice

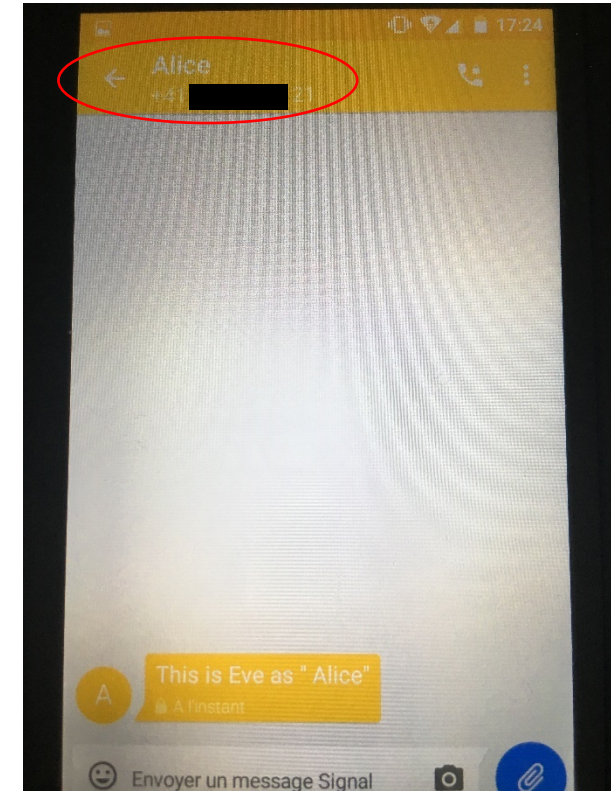


Bob



Nasty trick: contact with similar name

- Signal 5
- Conversation with Eve as « Alice »



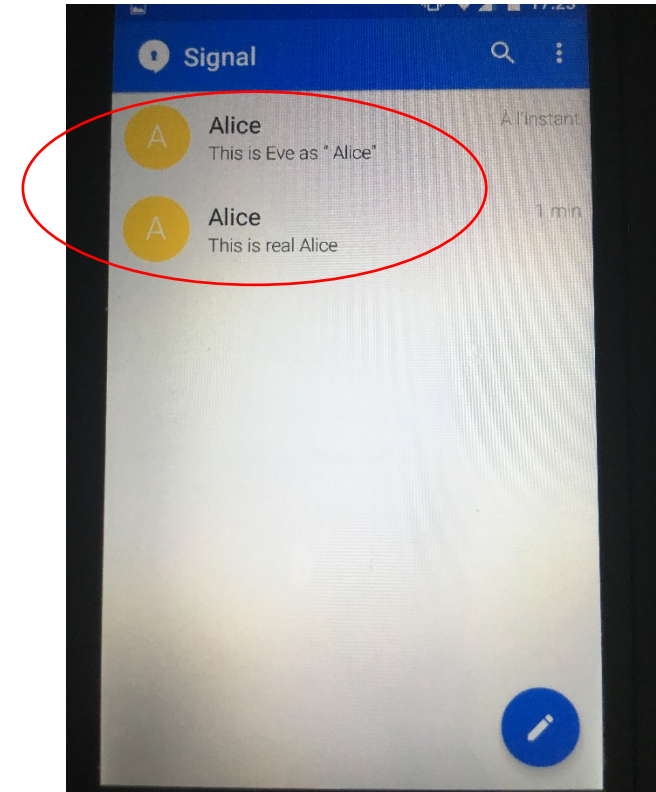
Bob



Nasty trick: contact with similar name

- Signal 3

- Main view



Bob



Contact with similar name results

- Creating « Alice » in addition to Alice is far more discrete
 - Phone call/SMS OK with real contact
 - Whitespace prefix is not visible in messaging apps
 - Requires a new contact, but MITC app can delete/recreate « Alice » as often as needed
- **Why does it work ?**
 - **Design error** from a security point of view: **phone number poor identifier**
 - **Abusing TOFU**: new contact = new key = accepted by default
 - **End user/mobile not really included in threat model**
 - Focus on protecting network/backend (e.g. against government agencies)
 - Side channel attack with some social engineering out of scope
- Yet after a few messages, Bob can guess it is not really Alice speaking to him

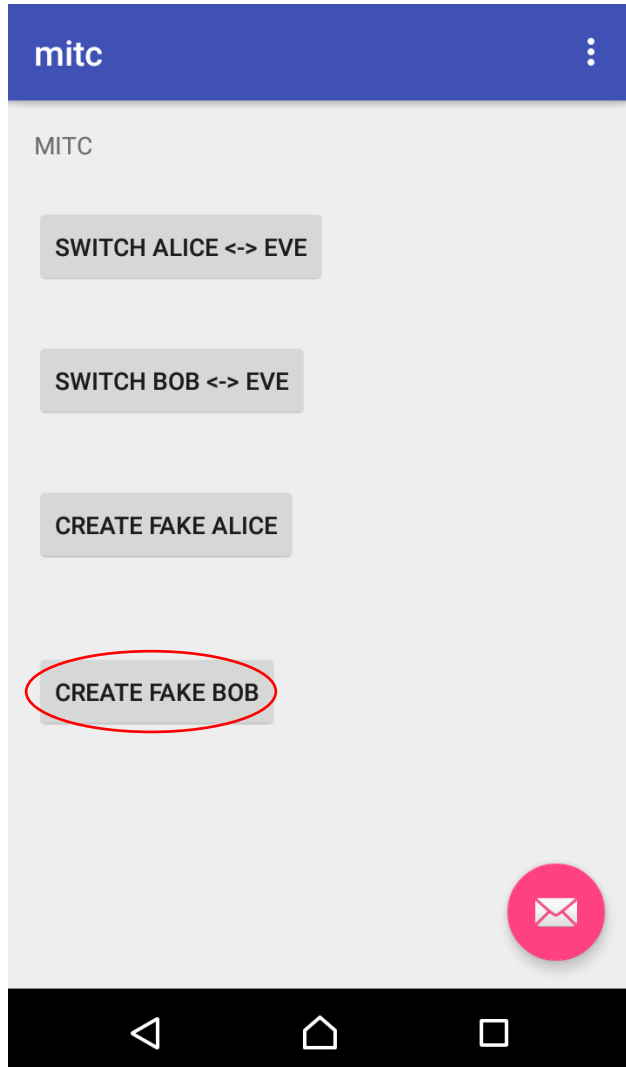


Building an attack scenario with MITC

- Let's build an easy exploitation scenario
- Convinced many more similar attacks are possible
 - Look in detail in the implementations how contacts are handled
 - Reverse engineering for WhatsApp/Telegram
 - Highly likely Java readable code for handling the Android contacts
 - Open source code for Signal
 - Extra identifiers stored in contacts: they can also be modified !
- **Suppose Alice also installed the MITC app**
 - because it's very popular
 - or MITC app sends her a SMS recommending to do so because it found her in Bob's contacts



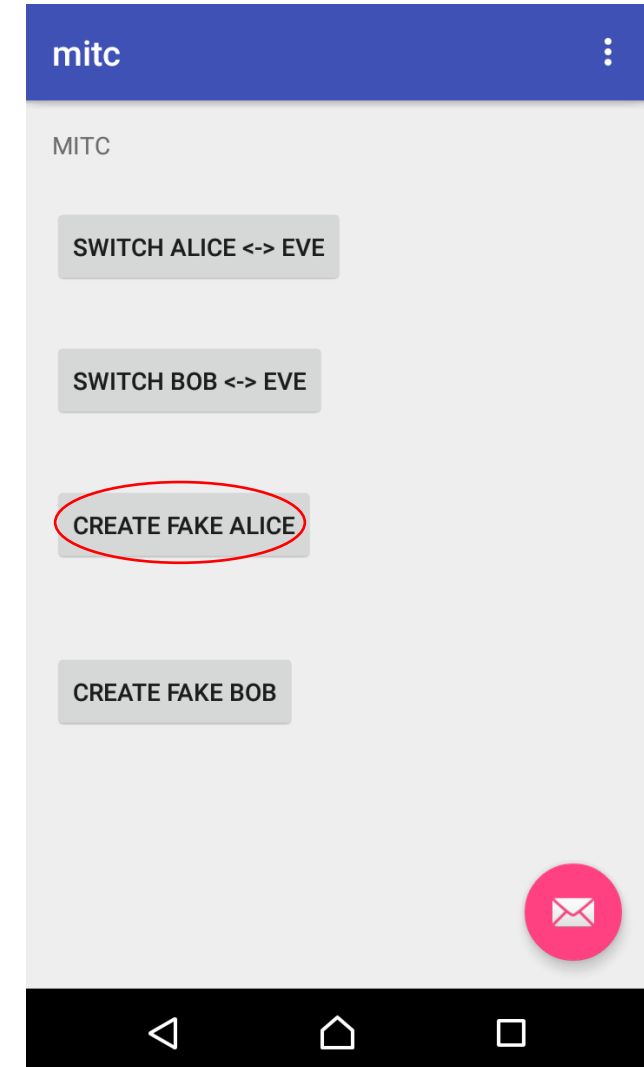
Man In The Middle: init phase



Alice

- 1. Have MITC deployed on the devices of Alice and Bob
- 2. Login as Eve to **web version of messaging app**
- 3. Create « Alice » and « Bob » with Eve's phone number via MITC app

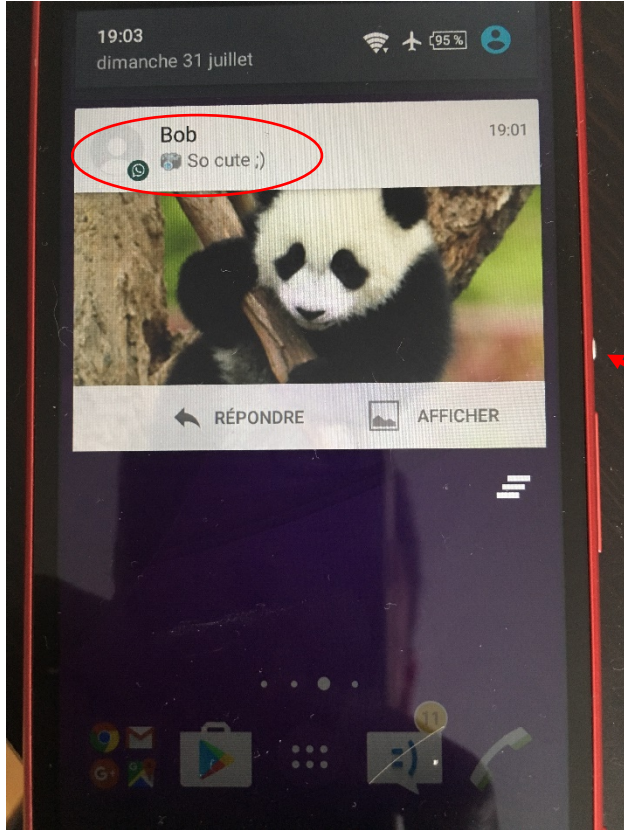
Eve



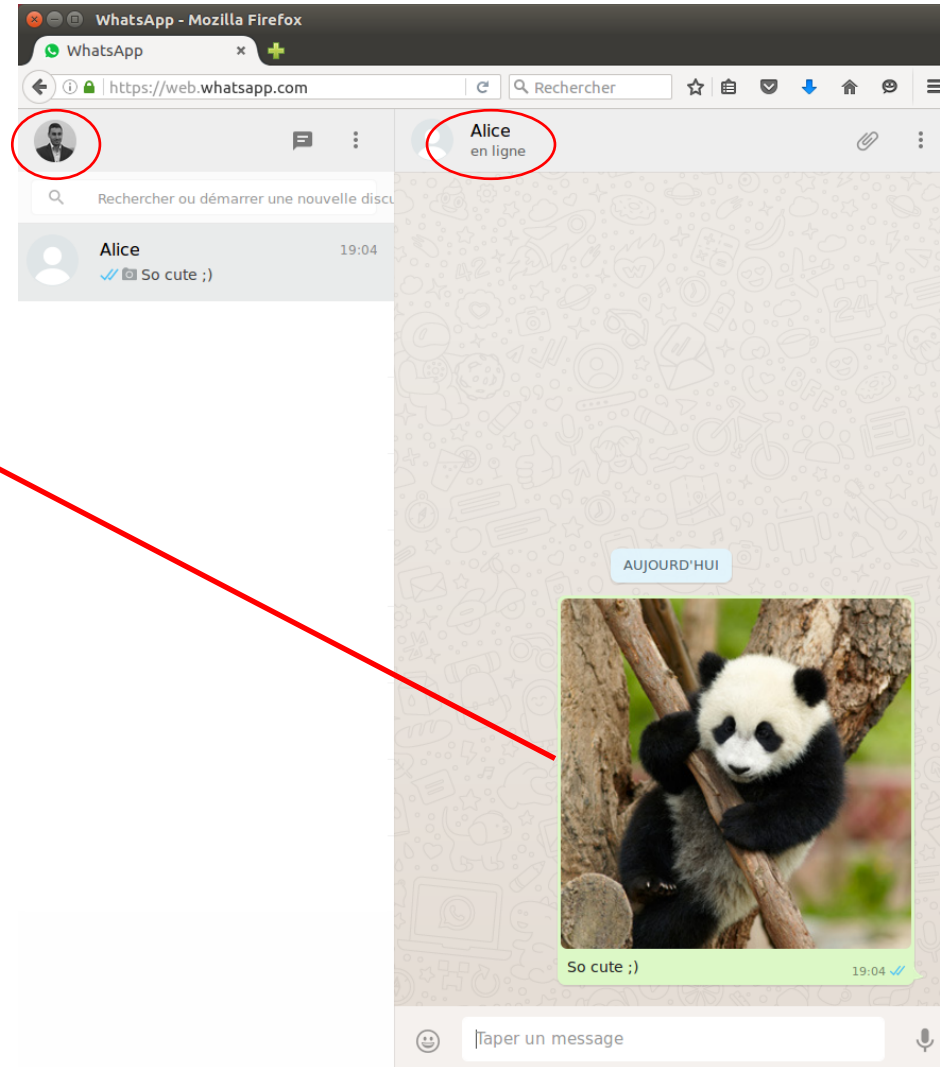
Bob



Man In The Middle: provoke discussion 1



Alice

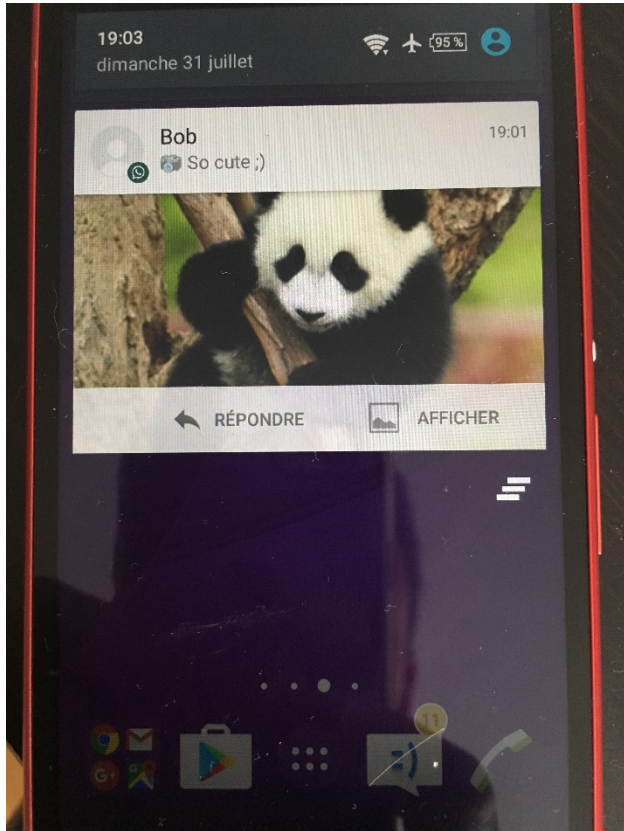


Eve

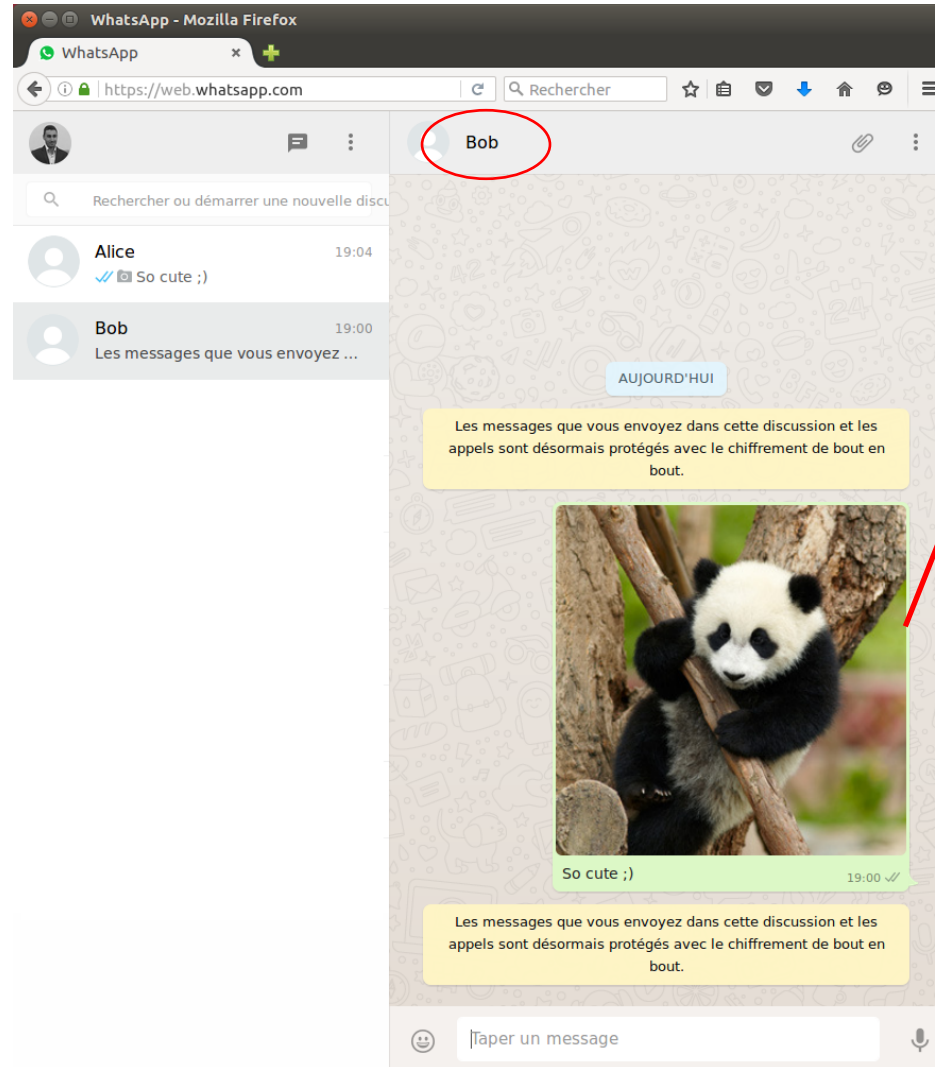
Bob



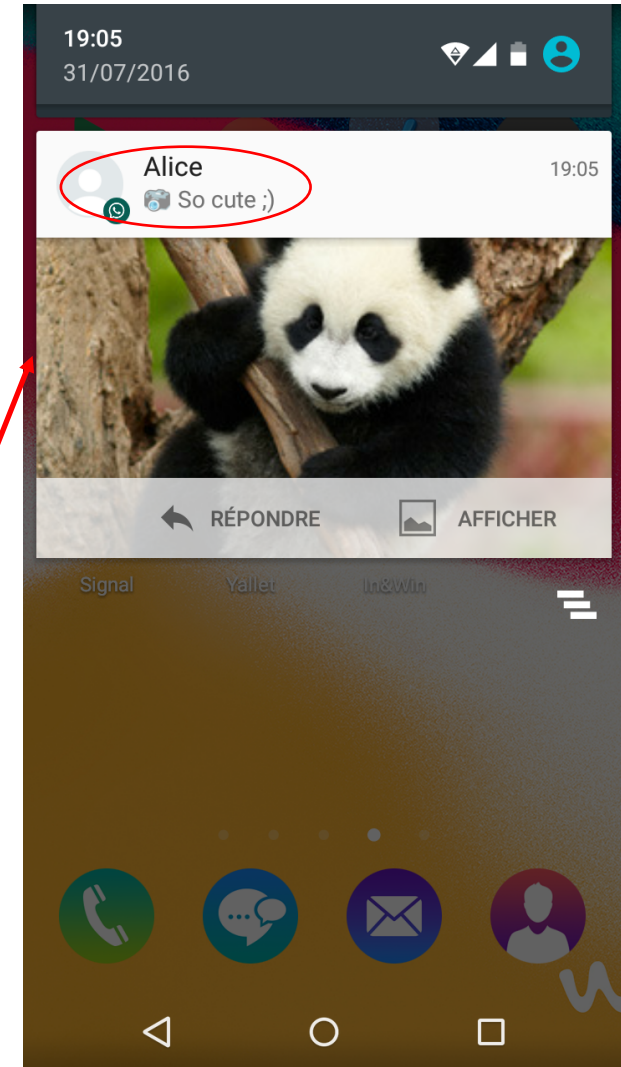
Man In The Middle: provoke discussion 2



Alice



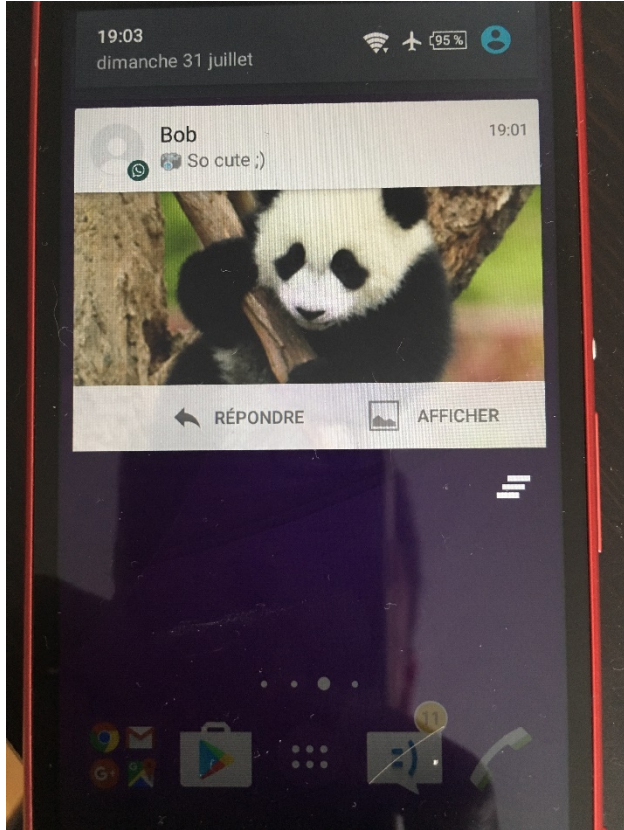
Eve



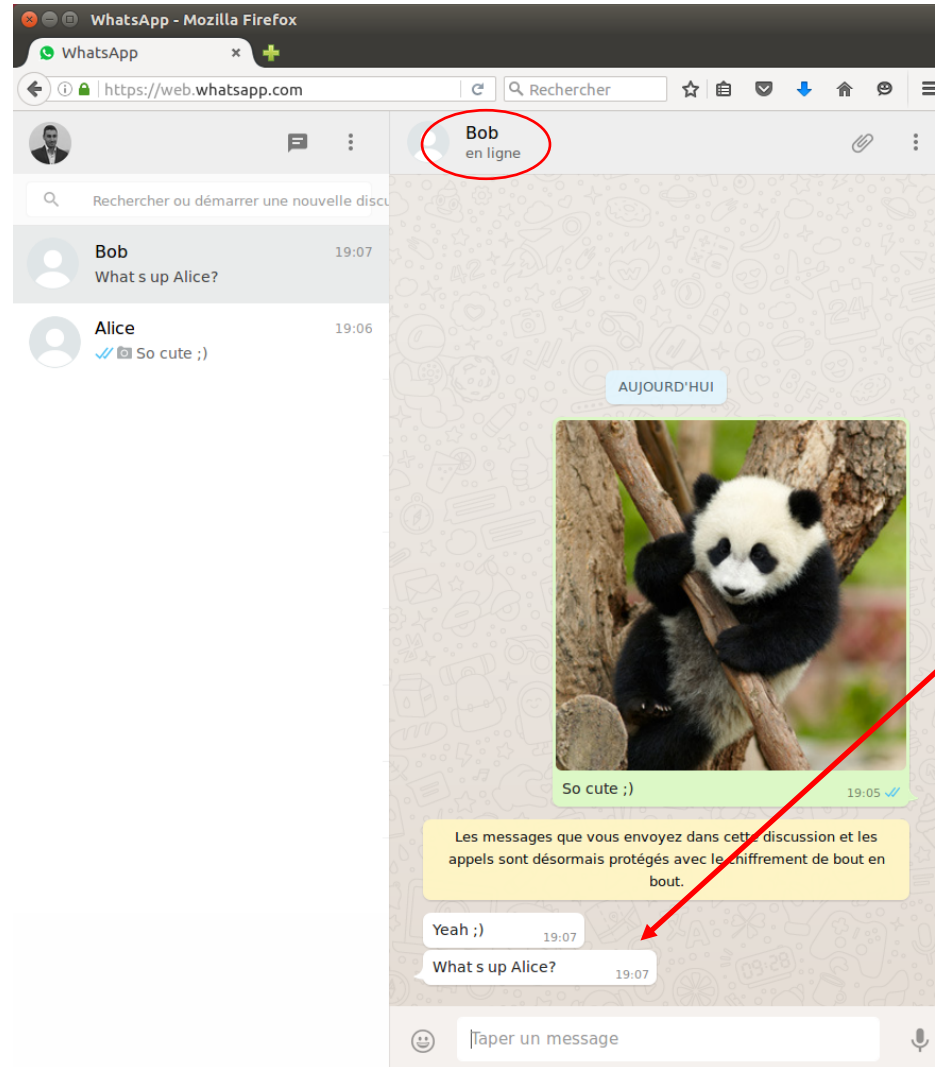
Bob



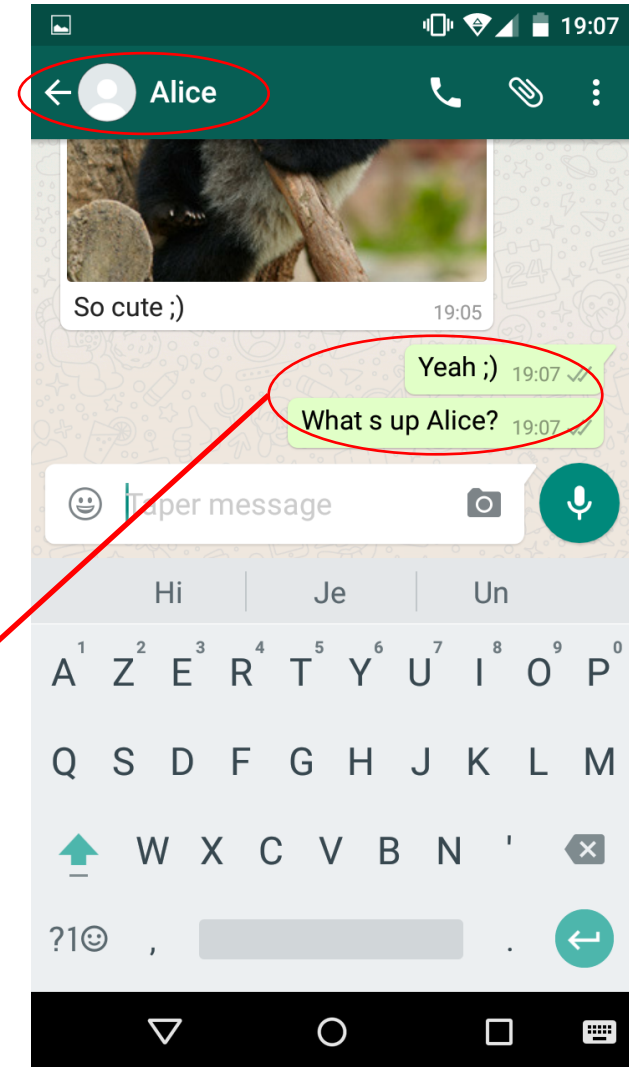
Man In The Middle: discussion 1



Alice



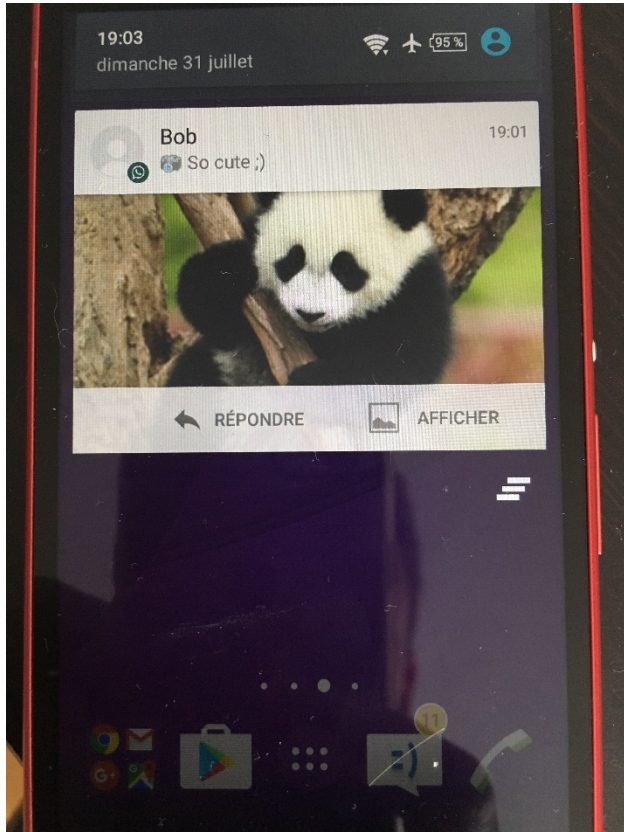
Eve



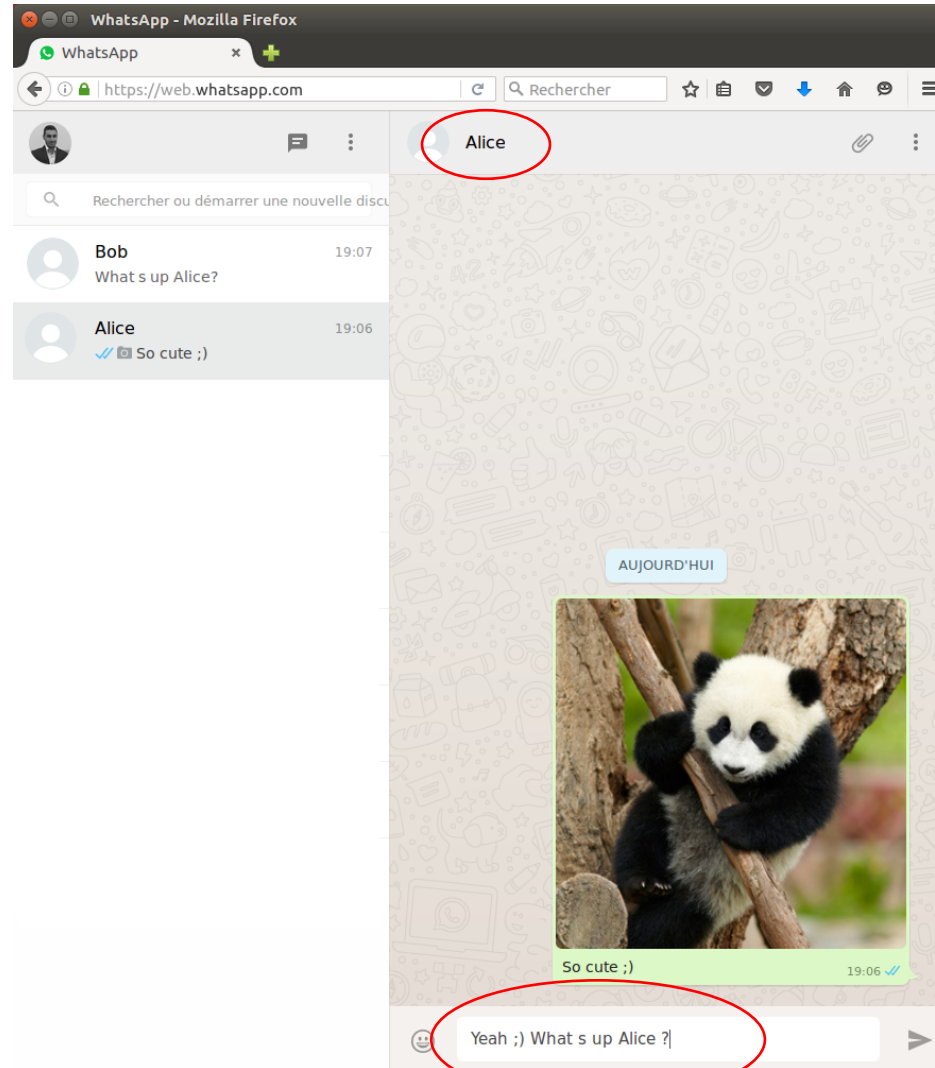
Bob



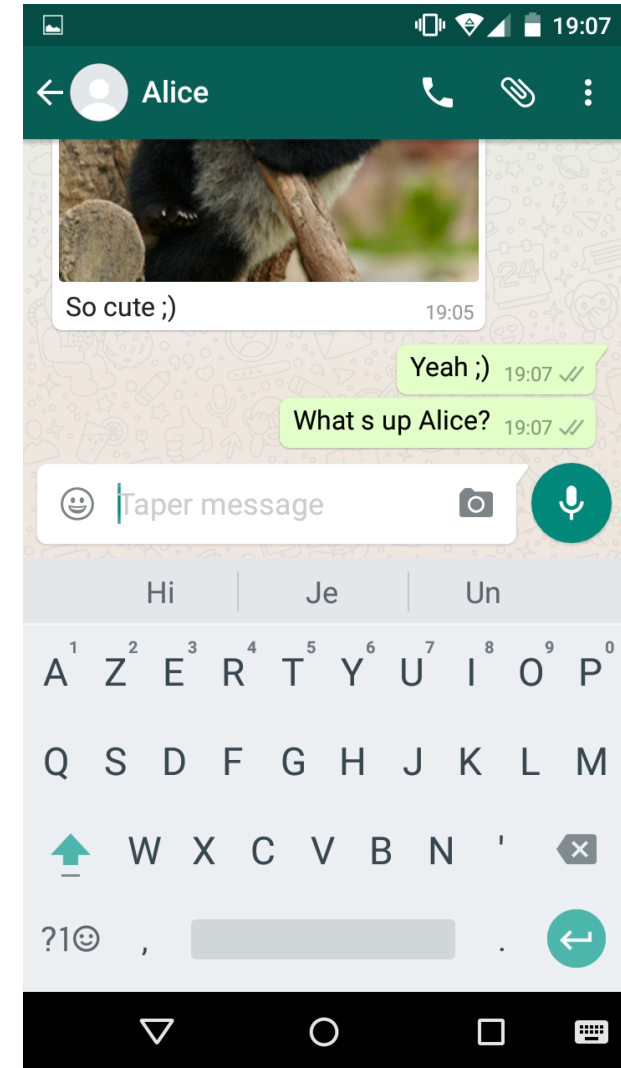
Man In The Middle: discussion 2



Alice



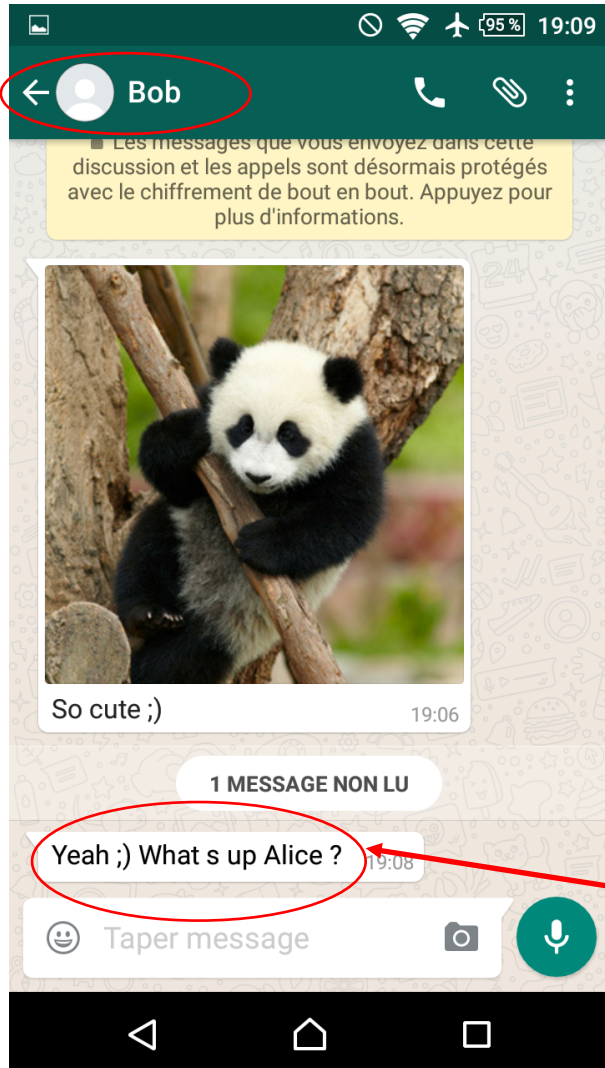
Eve



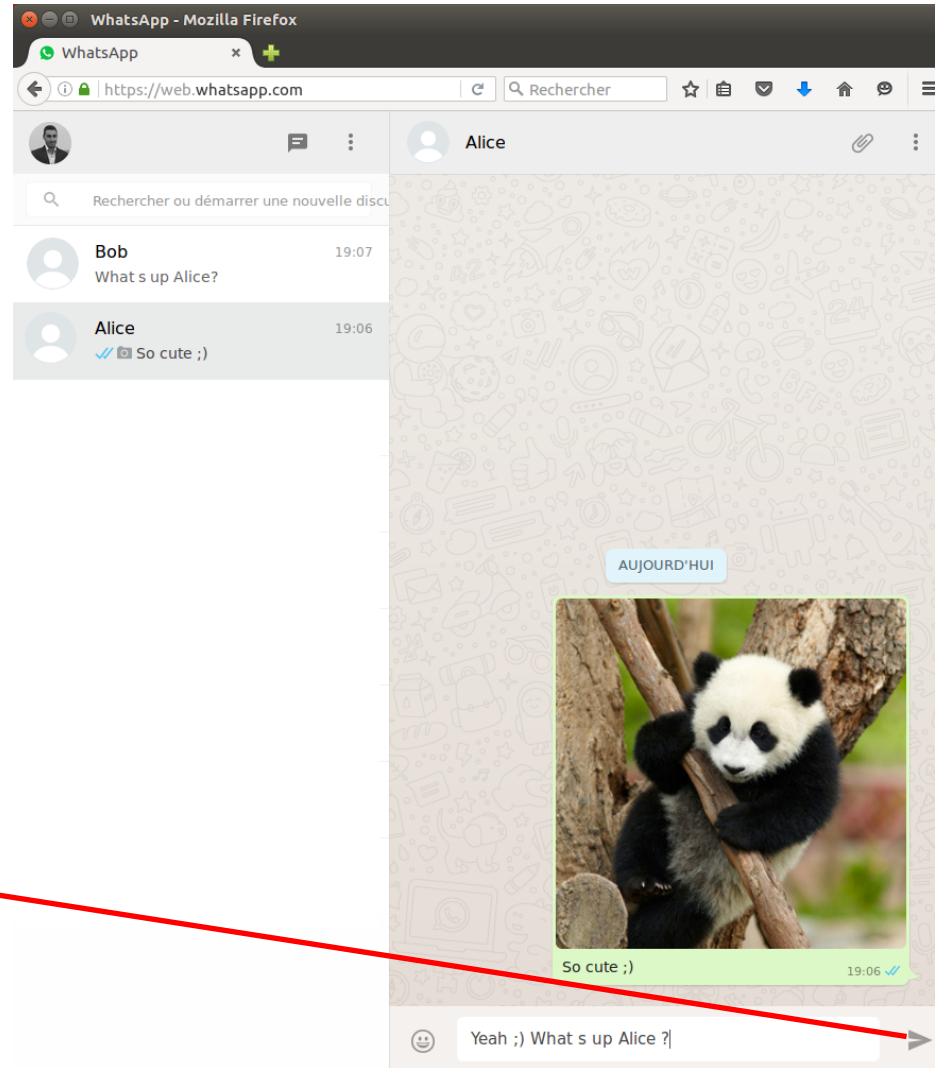
Bob



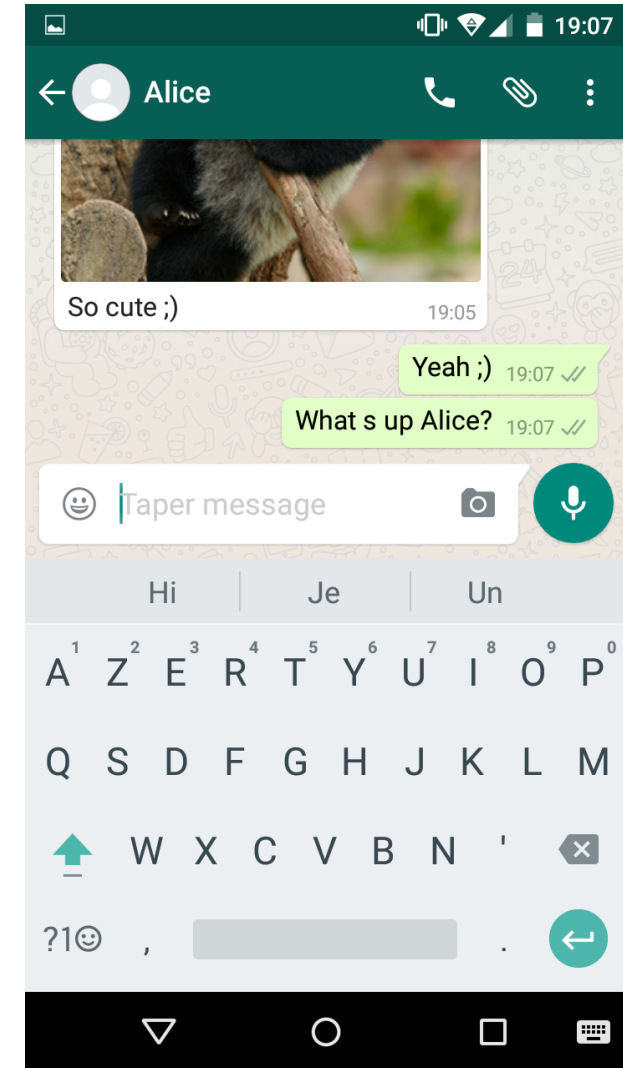
Man In The Middle: discussion 3



Alice



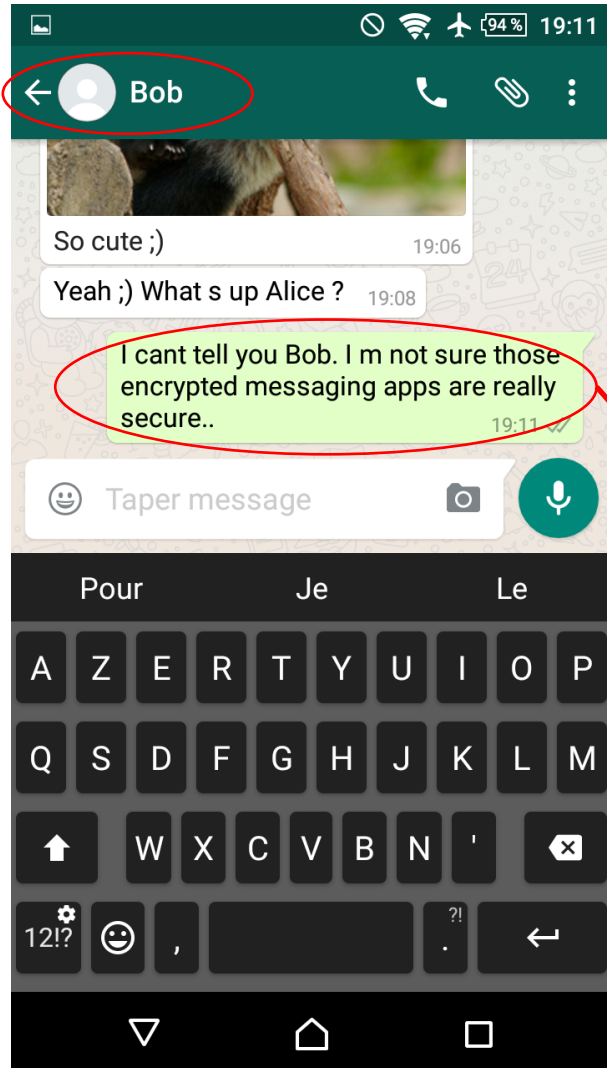
Eve



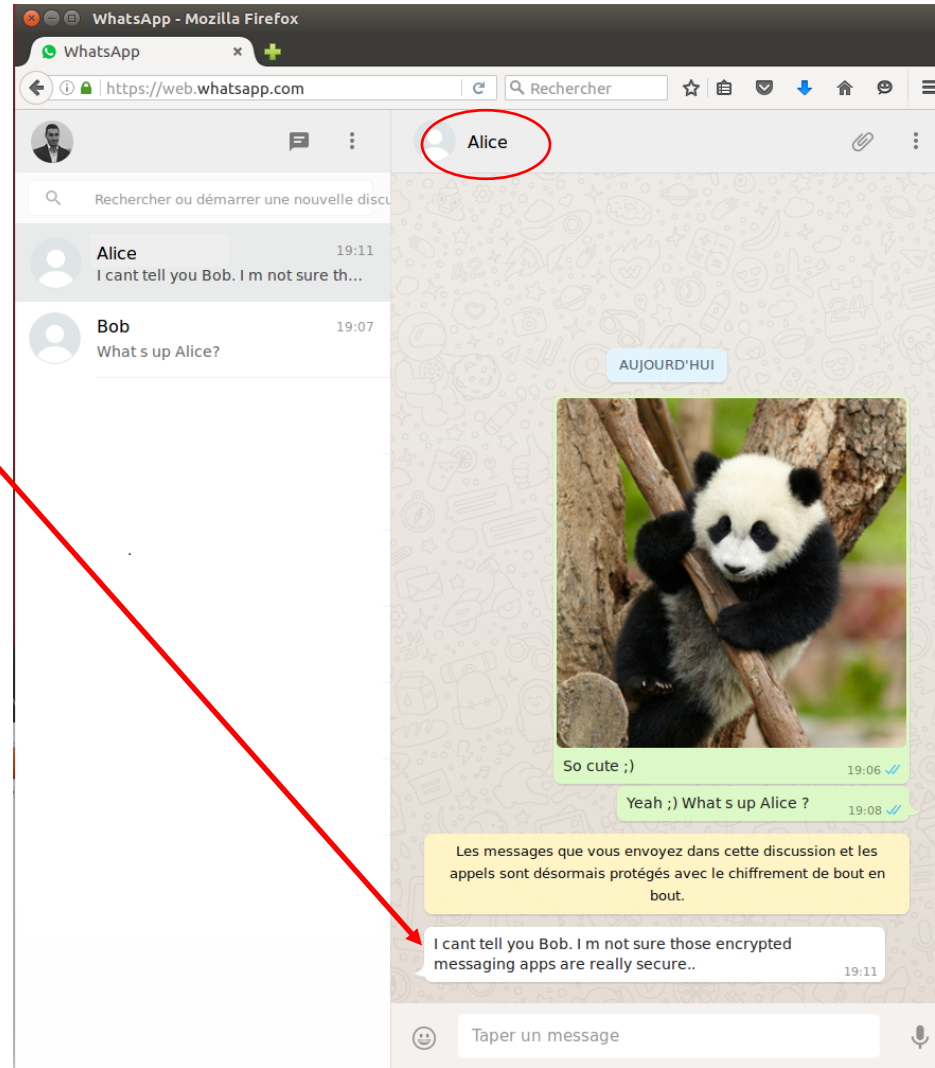
Bob



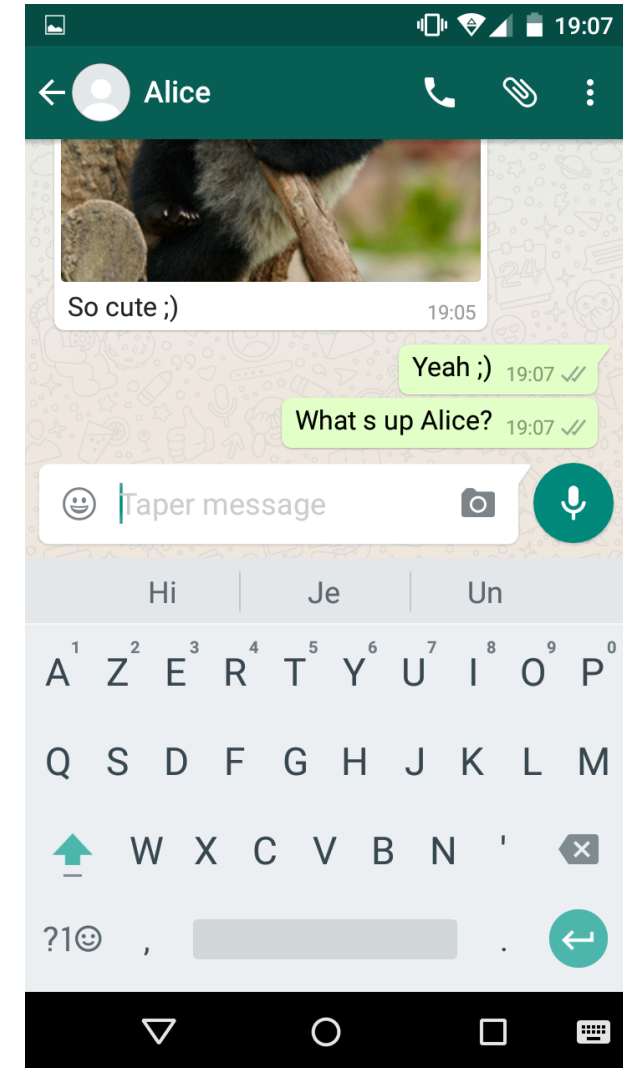
Man In The Middle: discussion 4



Alice



Eve



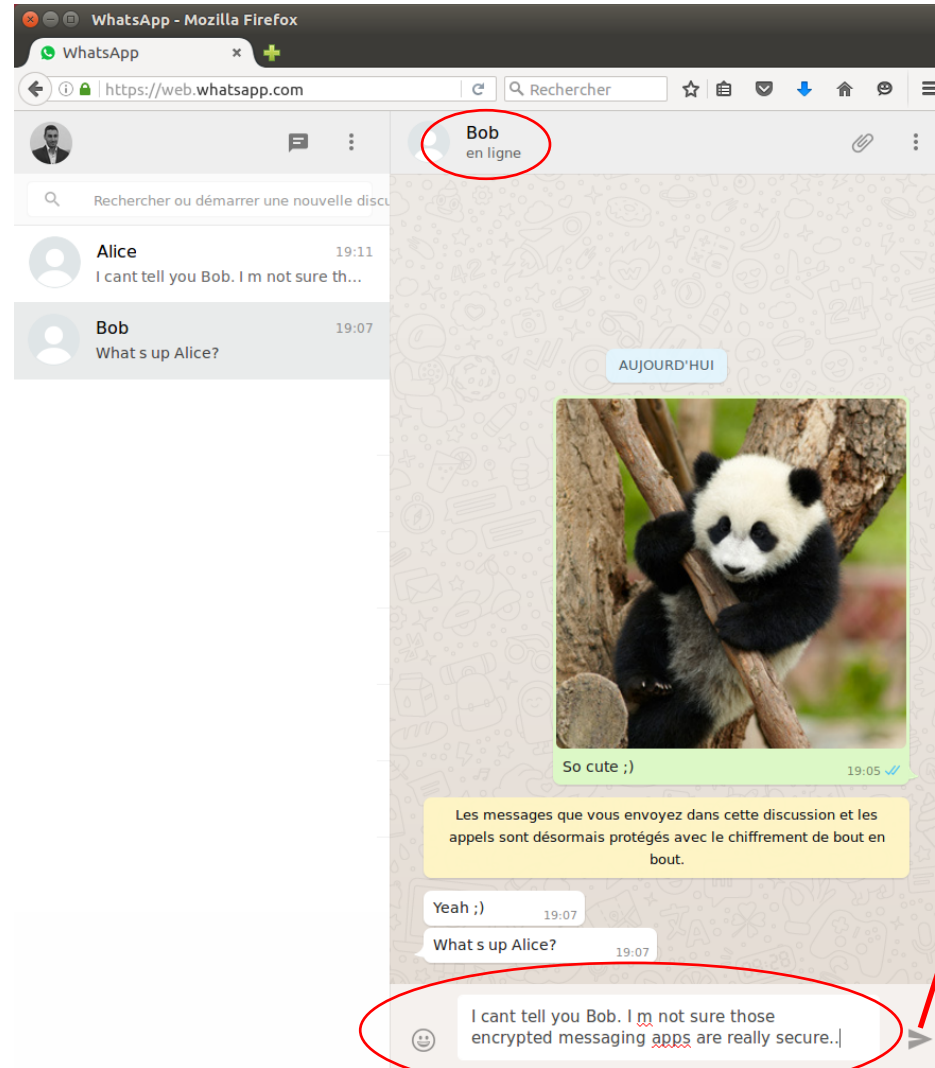
Bob



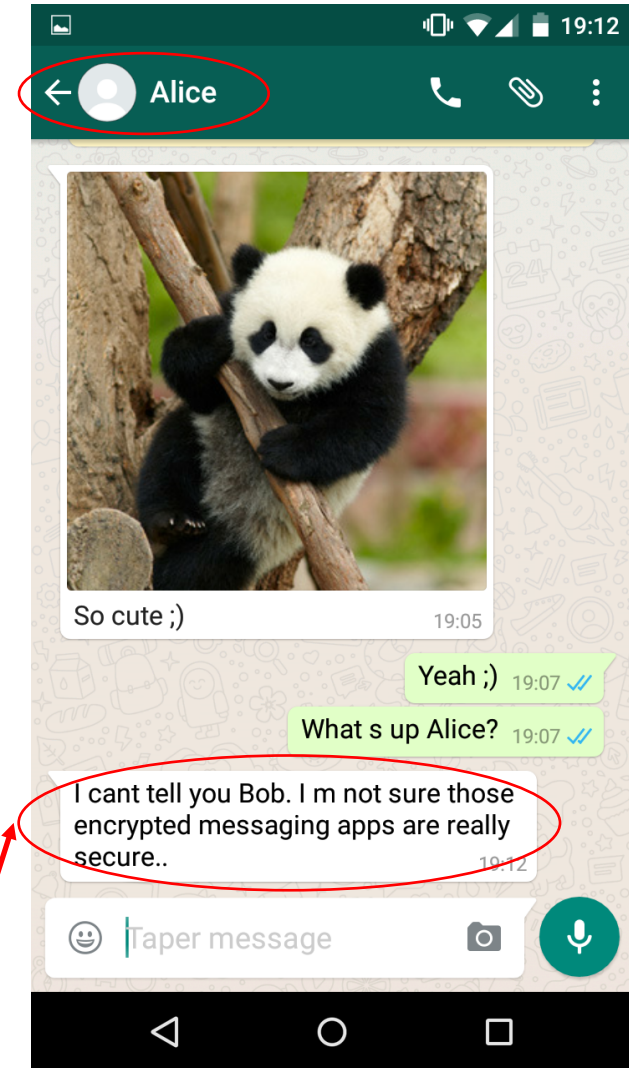
Man In The Middle: discussion 5



Alice



Eve



Bob



Man In The Middle results

- WhatsApp

- **Possible to share a real conversation between Bob and Alice via Eve**
- Only need to switch to a new conversation by forcing a chat
- Later conversations will likely continue in this session (UI easiest path)

- Telegram: same results (web version also available), as long as the new contacts are used for the first time

- Signal: same results

- Phone number always displayed below contact name (Android version only)



Risk assessment

- Simple evaluation: $\text{risk} = \text{easiness of attack} * \text{user impact}$
- Difficulty of attack: Low-Medium
 - Technically: Low
 - Easy to access contacts via code
 - Not a problem to get MITC application approved for publication
 - Logistics : Medium
 - One phone number is enough
 - Need to convince many users to install the MITC application
 - But « Ponzi scheme » possible by using the contact information
- Impact: High
 - Thousands of users can be spied: multi-app + multi-mobile os

Difficulty to attack	Low business impact	Medium business impact	High business impact
Low	Low	Medium	Very High
Medium	Low	Medium	High
High	Low	Low	Medium



Vendors feedback 1/4

●Telegram

- Very efficient Level 1 support (a contact in Telegram app)
- Level 2: security@telegram.org = **/dev/null**
 - Contacted them 3 times
 - Asked Level 1 to recontact them
 - Public question via Twitter

●WhatsApp

- Contact Facebook security via form => automatic confirmation
- No answer for one month
- Recontacted them: answer received the next day
- Replied to it, but never got feedback



Vendors feedback 2/4

●WhatsApp answer (layout as is, bold added)

We appreciate your report. **Ultimately** an attacker with **malware** installed on a device is going to be able to alter data on the device itself. In your examples for **WhatsApp conversations remained properly bound to the phone number that the messages were sent to**. Beyond that, WhatsApp allows people to **set local aliases for contacts** and to view the number associated with a specific message thread at any point. Given that, we don't feel that this behavior poses a significant risk and **we do not plan to make any changes here**. Please **let us know if you feel we've misunderstood something** here!



Vendors feedback 3/4

●Signal

- Not clear what is the channel for security issues
- Create report for Android app in Signal bug tracking tool
- Recontacted twice but no answer for 2 months
- Public question via Twitter
 - Quick answer from support
 - Someone from tech team will contact me soon
- After recontacting support, discussion started



Vendors feedback 4/4

●Signal 1st reply (layout as is, bold added)

Hey Jeremy, saw your support email about "man in the contacts." This, like all interception techniques, is **what safety numbers are for. Signal users would be notified that** the safety numbers for **their contact have changed**, and be asked to verify them. A successful MITM attack would need to find a way to intercept communication without triggering that notice.

●Signal 2nd reply (layout as is, bold added)

Hey Jeremy, **Signal is not designed to protect your device against malware.** Thanks for getting in touch, good luck with everything.



Countermeasures: mobile apps

- Give up the implicit trust on contacts
 - Deal with contacts the old way
 - Provide an explicit identifier + authenticifier
 - Manually approve contacts to be added
- Or mitigate risk by increasing user awareness
 - Clear message when a new conversation starts explaining to be cautious
 - Ideally warning sign in the corresponding notification
 - Display a visible trust level indicator next to the contact name
 - If no chat history for this number/key, trust level set to minimum
 - Phone number can help (if people know the numbers by heart ...)
- Mobile OS: stronger restrictions for accessing contacts



Countermeasures: end user

- Check your contacts
- Avoid installing applications asking for modify contacts permission
 - Yes messagins apps ask for this permission ...
- Use Threema
 - Swiss German app
 - Manual id handling
 - Optional contact sync
 - Visible trust level: Red/Orange/Green
 - Questions on contacts handling sent to press@threema.ch
 - Very detailed answer with the clear design choices received the next day



Conclusion

- **E2E can't guarantee privacy if you're not sure who you're talking to**
 - Beware of messages displaying good cryptography is used because it can bring a false sense of security
- **Security model around contacts is far too open** for sensitive apps
- **Authenticating the other party is an absolute necessity**
 - But it's a difficult task, particularly the provisioning processes
 - And even more to make it user friendly
 - **The end user must be in the loop** to detect suspicious activity
 - If it's too complex, secure features won't be used
 - A significant part of end users will install crappy apps, accept anything and not care about security warnings
- If the design of your solution includes access to contacts, start a threat modeling session



Thank you !



Any question ?

