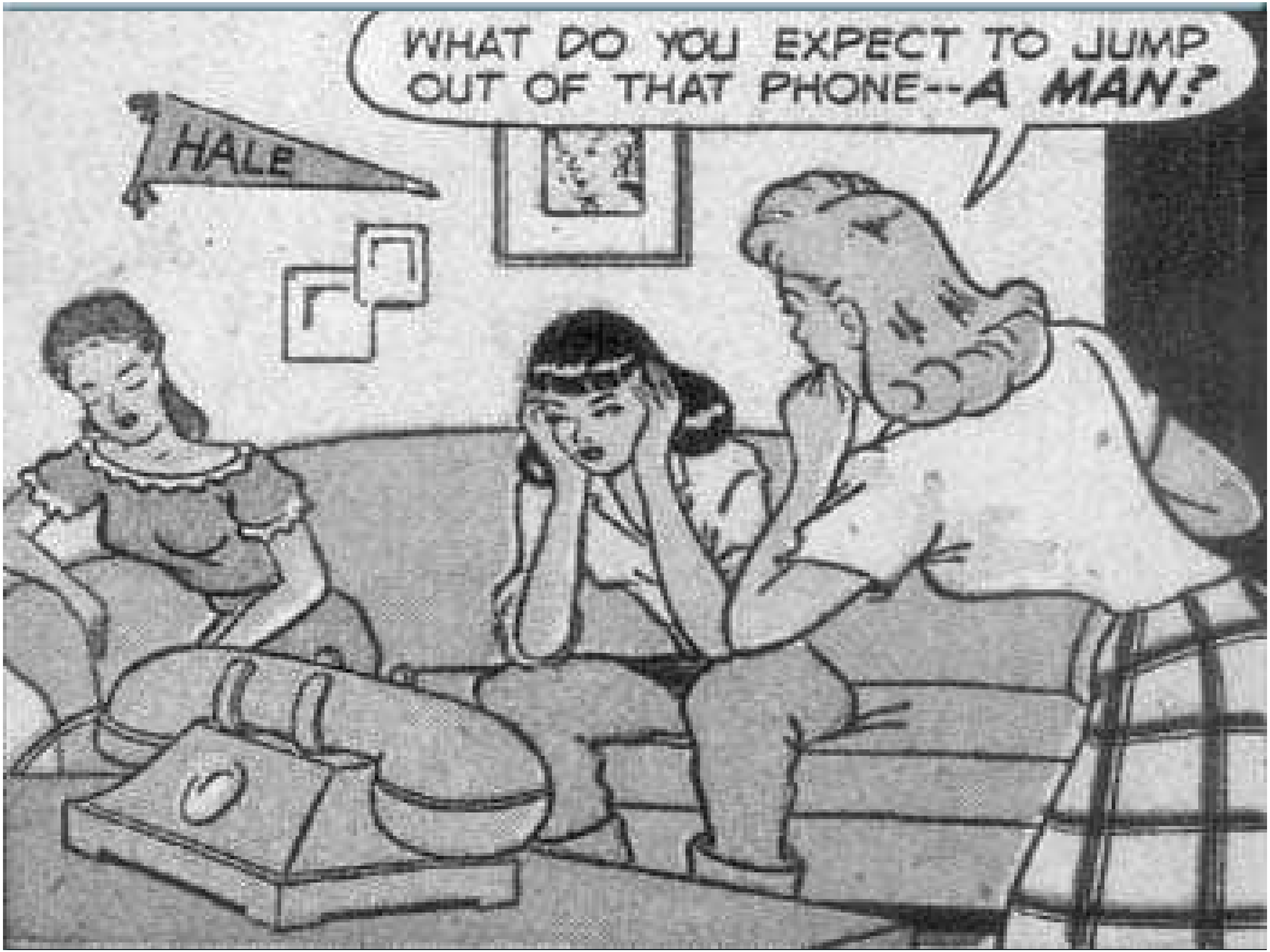
Unregister Attack in SIP

Anat Bremler-Barr Ronit Halachmi-Bekel Jussi Kangasharju

Interdisciplinary center Herzliya *Darmstadt University of Technology*

Unregister Attack

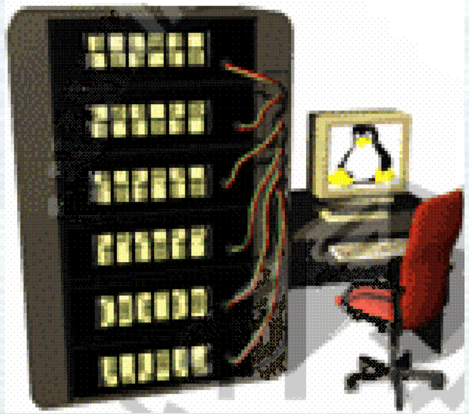
- We present a new VoIP Denial Of Service/impersonating attack
- Attacker cancels the registration of the phone number in the system
- The victim can no longer be reached
- The victim has no idea that he cannot be reached


HALE
WHAT DO YOU EXPECT TO JUMP OUT OF THAT PHONE--*A MAN?*

- Introduction to Telephony
- VoIP
- SIP
- Unregister Attack
- SOHA Solution
- Conclusion

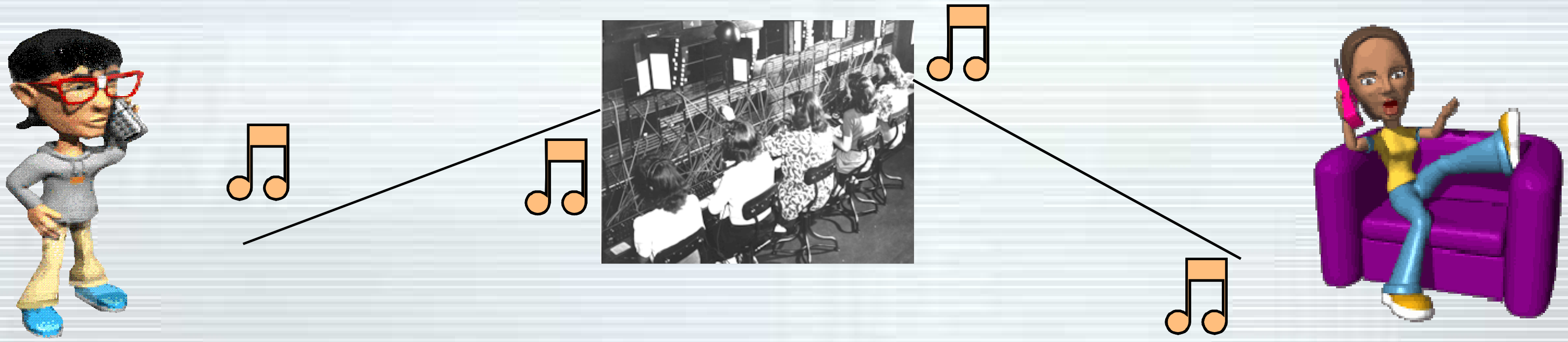
Introduction to Telephony

- Manual switchboards
- Electronic switchboards
- VoIP phones
- The technology is still changing dramatically



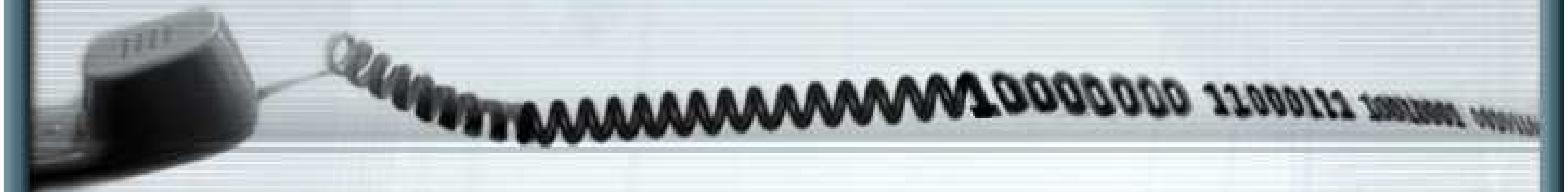
Circuit Switching

- Two sides of the call creates an electric circuit between them.
- All the communication of the call travels this circuit, and the channels are fully dedicated to the call
- Waste of resources when there is silence
- Designated specifically to phone calls
- Considered secure



Packet Switching

- No physical link between source and destination, the path between them varies
- Resources are shared by all users (Internet)
- No meaning to physical location
- Vulnerable



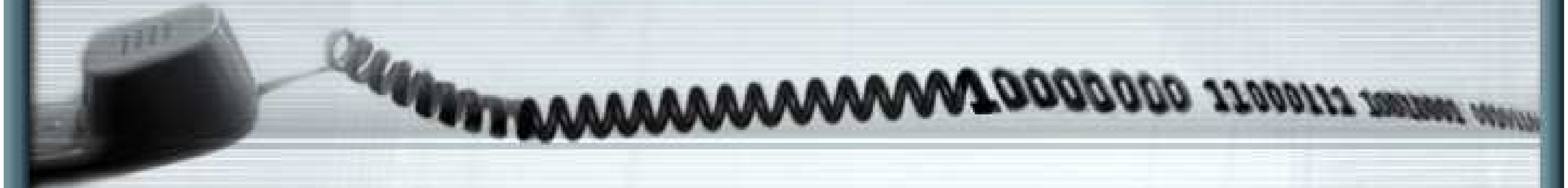
- Introduction to Telephony
- VoIP
- SIP
- Unregister Attack
- SOHA Solution
- Conclusion

Voice Over Internet Protocol (VoIP)

- A technology that allows phone calls to be made over the internet
- Packet switching
- Use of existing technology
- Two Phases:
  - Registration (dynamic / static)
  - Calls
- Signaling: SIP, MGCP, H323
- Media: RTP

VoIP Advantages

- Resources are shared by all users.
- Mobility
- Functionality
  - Forking
  - Advanced call flows
- Cost
  - Uses existing platforms
  - Decreased price of domestic and international calls



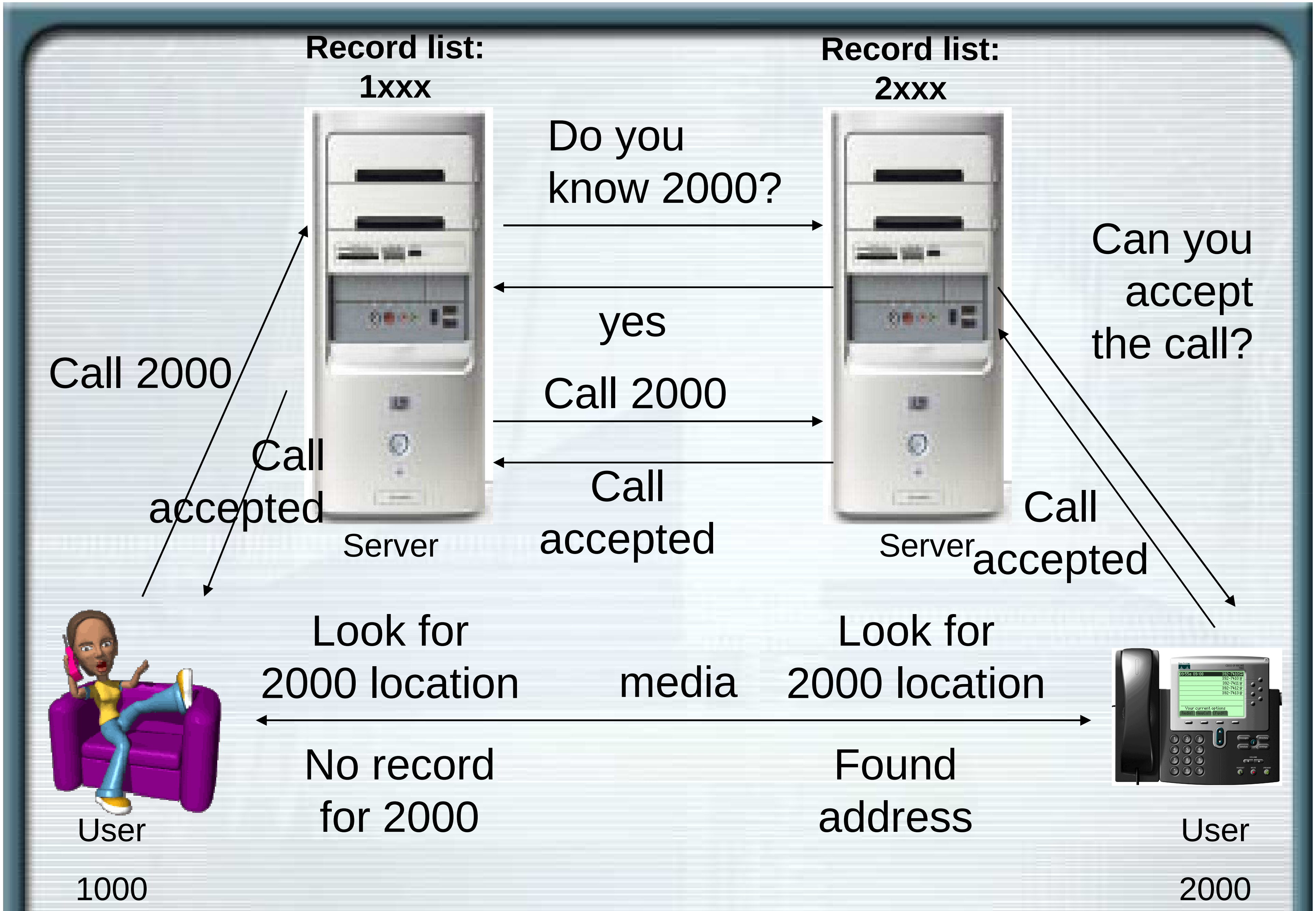
- Introduction to Telephony
- VoIP
- SIP
- Unregister Attack
- SOHA Solution
- Conclusion

Session Initiation Protocol (SIP)

- A text based signaling protocol
- Works mostly over UDP
- Used for internet telephony, instant messaging and presence services

SIP (Cont.)

- Client-server model
  - Client: telephone (endpoint)
  - Server
- Messages
  - Request
  - Responses



Record list: 1xxx

Record list: 2xxx

Do you know 2000?

yes

Call 2000

Call accepted

Can you accept the call?

Call accepted

Call 2000

Call accepted

Server

Server

Look for 2000 location

media

Look for 2000 location

No record for 2000

Found address

User

1000

User

2000

Registration

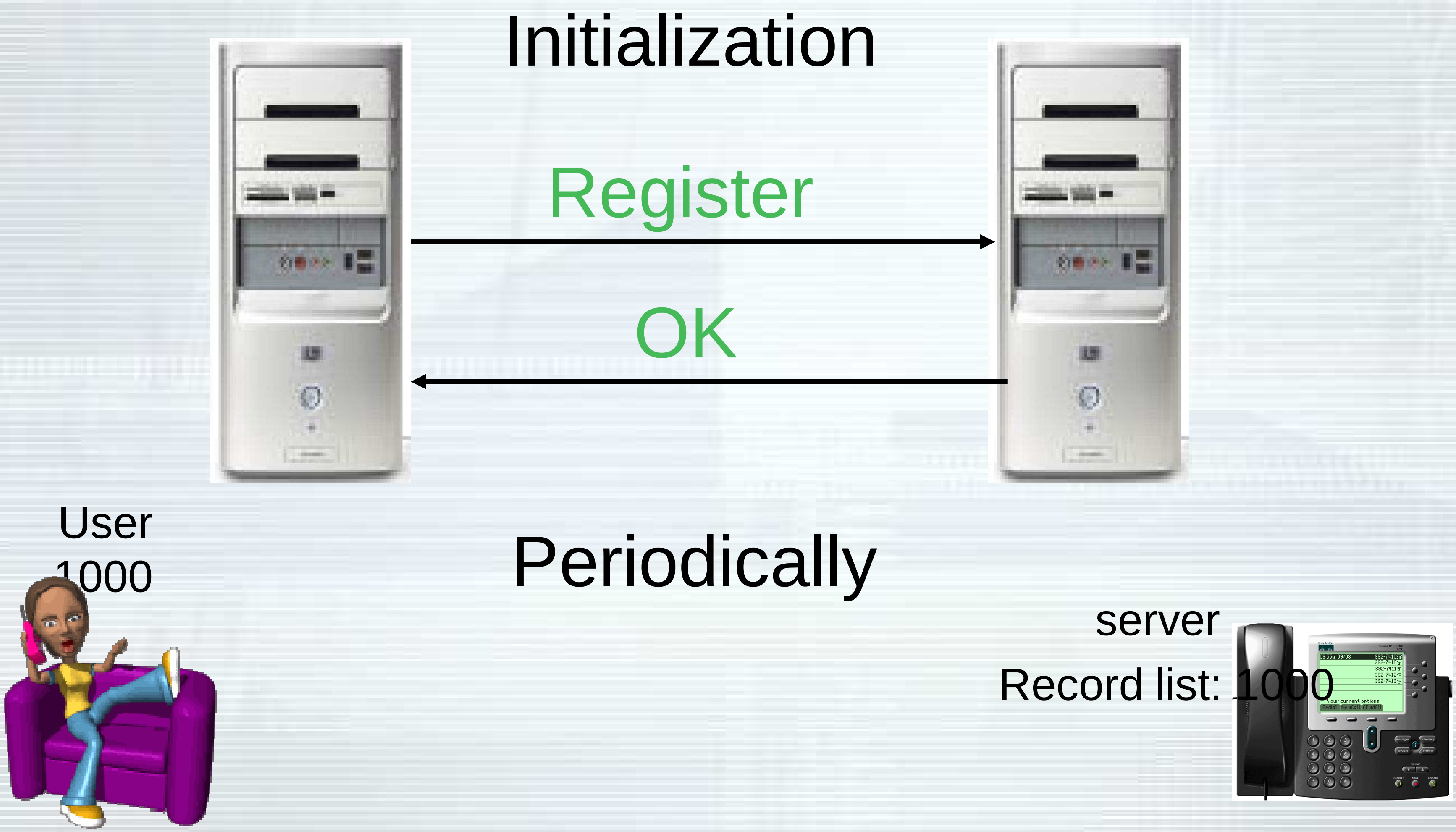
- Set of messages that eventually forms a record at the server
- Record: a foursome of the type
  Name, Number, IP, Port
- Authorization and authentication are possible:
  - Handled per request
  - A challenge/response mechanism
  - The server SHOULD authenticate the endpoint (RFC 3261)

Registration (Cont.)

- Expiry field - indicates how long the record will be valid for

- Call-id – unique identifier that groups together a series of messages.
  - It MUST be the same for all requests and responses related to the same dialog.
  - It SHOULD be the same in each registration.
- Cseq - identify and order transactions

Registration server without authentication



Initialization

Register

OK

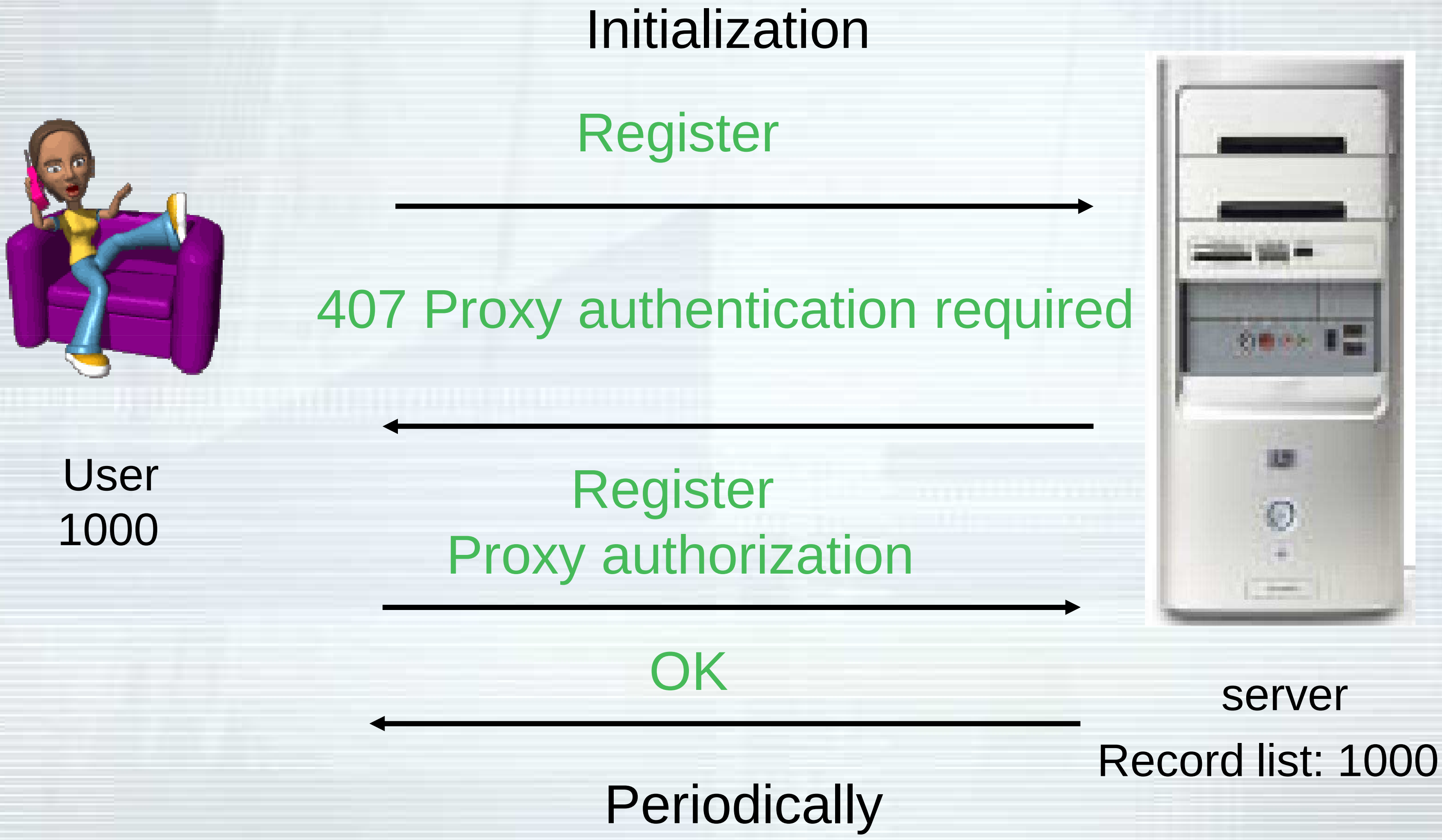
User 1000

Periodically

server

Record list: 1000

Registration server with authentication



Initialization

Register

407 Proxy authentication required

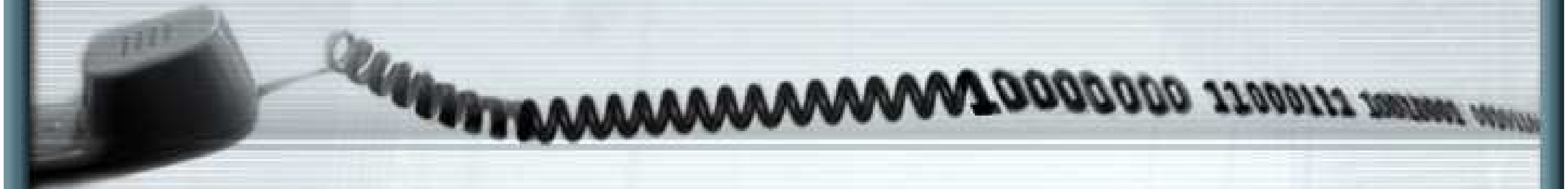
Register
Proxy authorization

OK

Periodically

Endpoint removal

- Endpoint removal:
  - Record expires according to the expiry value
  - User sent unregister message – a register message with expiry value of **zero**.
- Server SHOULD support the unregister message (RFC 3261).



- Introduction to Telephony
- VoIP
- SIP
- Unregister Attack
- SOHA Solution
- Conclusion

The Unregister Attack

- A new kind of Denial of Service/impersonating attack on SIP servers
- The attacker sends a spoof unregister packet
- As a result the server removes the victim’s record
- The victim has no indication that he is not registered at the server

Experiments

- The attacker uses a simple script written in C
- Tested on 3 different common servers
- Servers with/without authentication
- Attacker with/without traffic knowledge

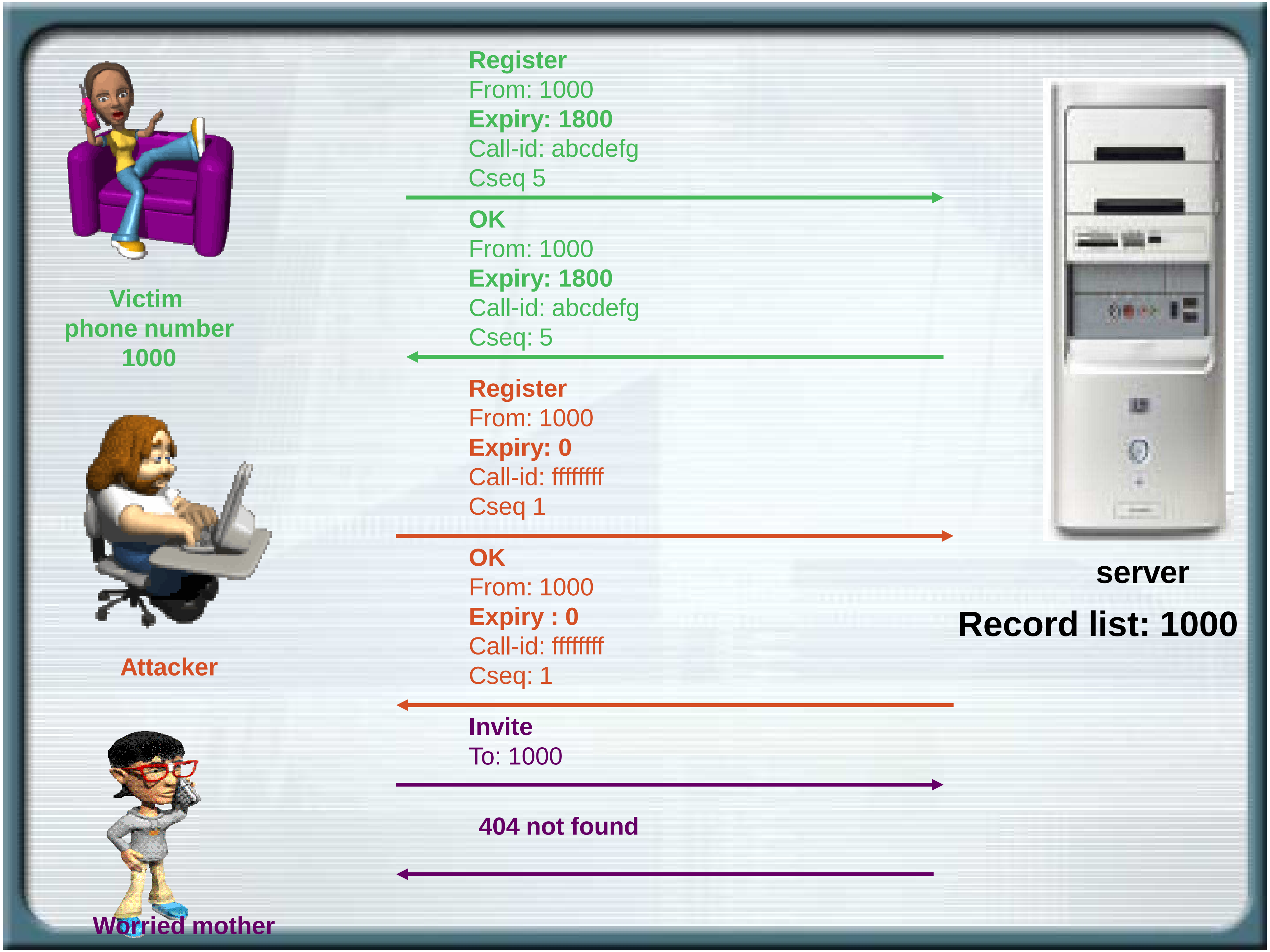
	Attacker without traffic knowledge	Attacker with traffic knowledge
Server without authentication	🟧	🟧
	🟦	🟦
Server with authentication	🟧	🟧
	🟦	🟦

Traffic knowledge – possible to receive

- Frustrated employee scenario
- Wireless
  - Public services without authentication
  - Some of wireless encryption is possible to decrypt

Server **without** authentication Attacker **without** traffic knowledge

- Pre knowledge:
  - IP address of the victim
  - Phone number of the victim
  - IP address of the server
- Prevention:
  - Verification *call-id* and *cseq* fields
- In practice the attack succeeded on two different common servers



Victim
phone number
1000

Register
From: 1000
Expiry: 1800
Call-id: abcdefg
Cseq 5

OK
From: 1000
Expiry: 1800
Call-id: abcdefg
Cseq: 5

Attacker

Register
From: 1000
Expiry: 0
Call-id: ffffffff
Cseq 1

OK
From: 1000
Expiry : 0
Call-id: ffffffff
Cseq: 1

Worried mother

Invite
To: 1000

404 not found

server

Record list: 1000

Attacks

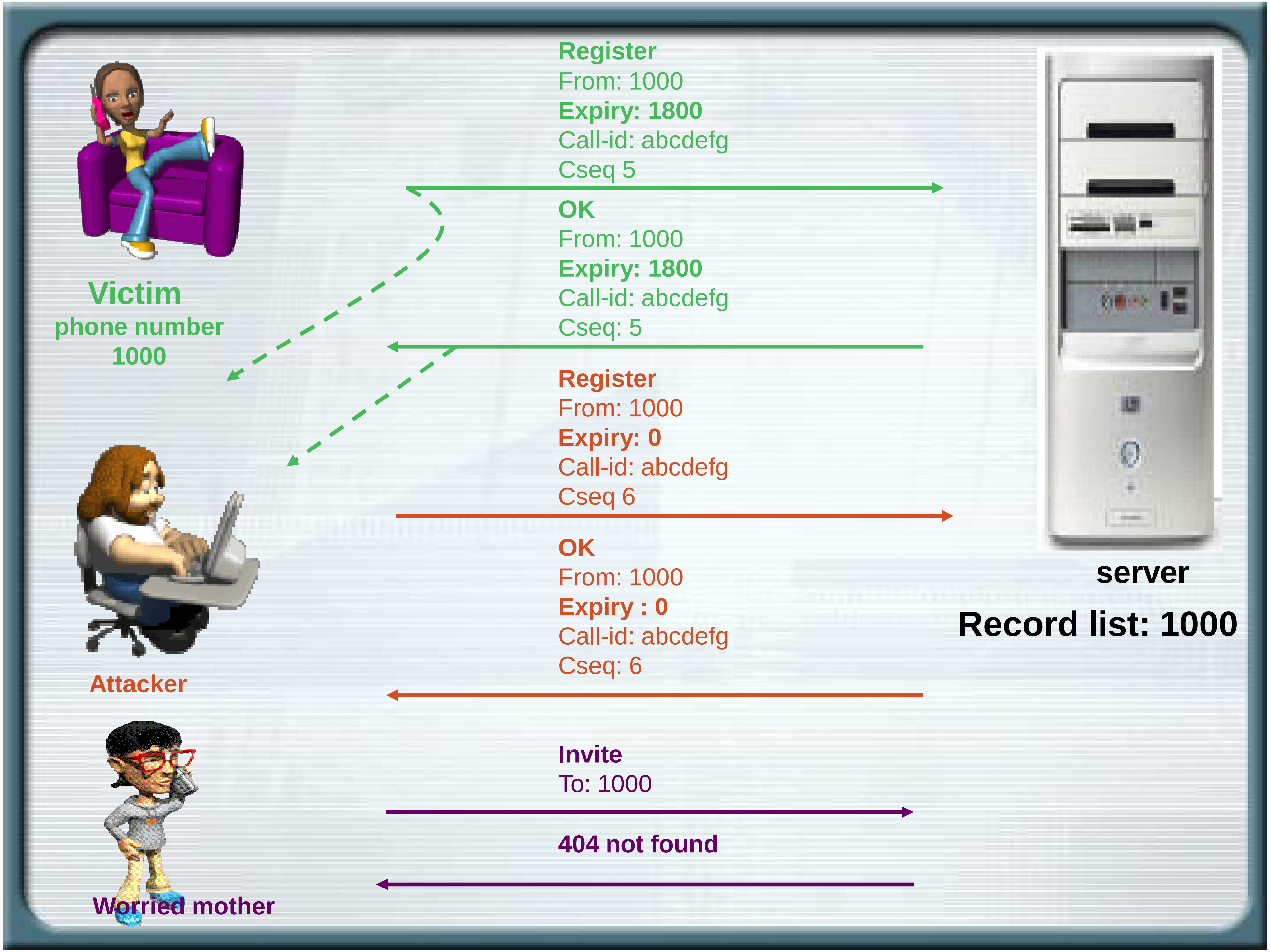
	Attacker without traffic knowledge	**Attacker with traffic knowledge**
Server without authentication	Permissive definition	
	call – id and cseq verification	
Server with authentication		

Problem

Solution

Server **without** authentication Attacker **with** traffic knowledge

- Pre knowledge:
  - IP address of the victim
  - Phone number of the victim
  - IP address of the server
  - Call-id and Cseq
- Without encryption of the packet there is no way to prevent the attack



Victim

phone number 1000

Attacker

Worried mother

Register
From: 1000
Expiry: 1800
Call-id: abcdefg
Cseq 5

OK
From: 1000
Expiry: 1800
Call-id: abcdefg
Cseq: 5

Register
From: 1000
Expiry: 0
Call-id: abcdefg
Cseq 6

OK
From: 1000
Expiry : 0
Call-id: abcdefg
Cseq: 6

Invite
To: 1000

404 not found

server

Record list: 1000

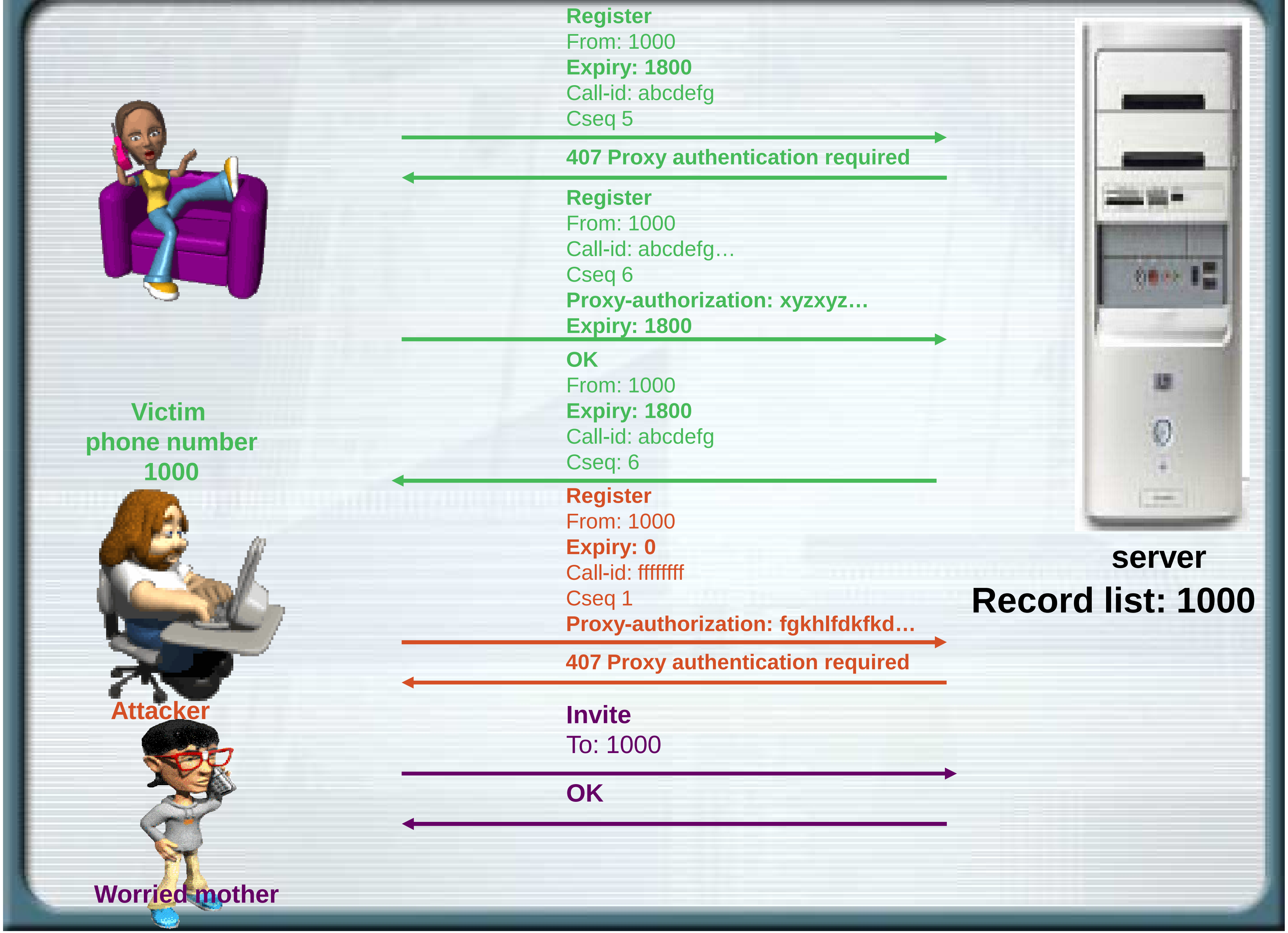
Attacks

	Attacker without traffic knowledge	Attacker with traffic knowledge
Server without authentication	Permissive definition	Not covered in the RFC
	call – id and cseq verification	SOHA solution (our solution)
Server with authentication		

Problem Solution

Server **with** authentication Attacker **without** traffic knowledge

- No way to perform the attack



Register
From: 1000
Expiry: 1800
Call-id: abcdefg
Cseq 5

407 Proxy authentication required

Register
From: 1000
Call-id: abcdefg…
Cseq 6
Proxy-authorization: xyzxyz…
Expiry: 1800

OK
From: 1000
Expiry: 1800
Call-id: abcdefg
Cseq: 6

Register
From: 1000
Expiry: 0
Call-id: ffffffff
Cseq 1
Proxy-authorization: fgkhlfdkfkd…

407 Proxy authentication required

Invite
To: 1000

OK

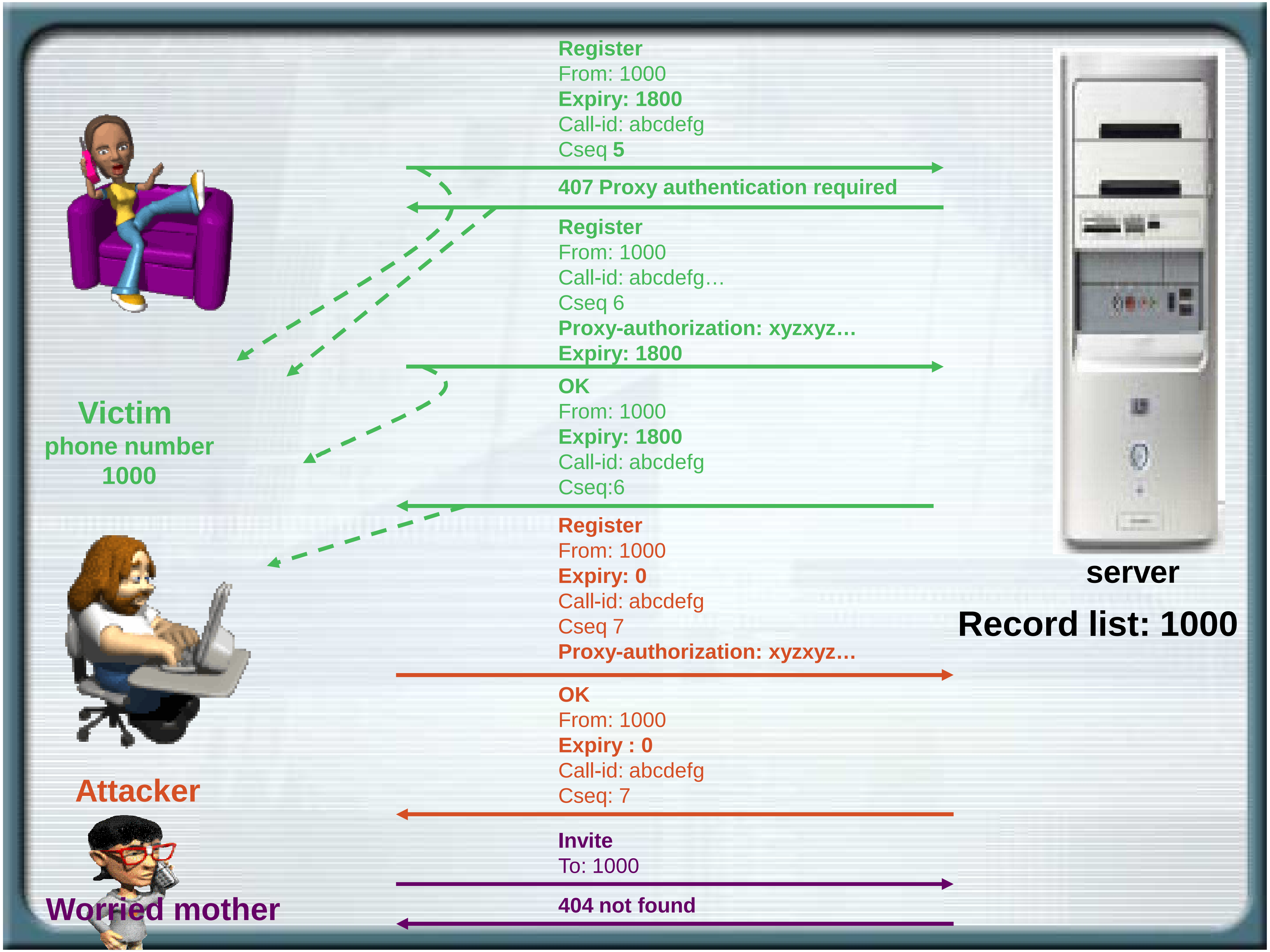
Attacks

	Attacker without traffic knowledge	**Attacker with traffic knowledge**
Server without authentication	Permissive definition	Not covered in the RFC
	call – id and cseq verification	SOHA solution (our solution)
Server with authentication	Impossible to attack	

Problem Solution

Server **with** authentication Attacker **with** traffic knowledge

- Replay attack –
  - Attacker uses the authentication data from the captured packet
- Pre knowledge:
  - IP address of the victim
  - Phone number of the victim
  - IP address of the server
  - Call-id and Cseq
  - **Previous** authorization field value
- Prevention
  - Server should ask for new authorization value for every packet it receives



Register
From: 1000
Expiry: 1800
Call-id: abcdefg
Cseq **5**

407 Proxy authentication required

Register
From: 1000
Call-id: abcdefg…
Cseq 6
Proxy-authorization: xyzxyz…
Expiry: 1800

OK
From: 1000
Expiry: 1800
Call-id: abcdefg
Cseq:6

Register
From: 1000
Expiry: 0
Call-id: abcdefg
Cseq 7
Proxy-authorization: xyzxyz…

OK
From: 1000
Expiry : 0
Call-id: abcdefg
Cseq: 7

Invite
To: 1000

404 not found

Victim
phone number
1000

Attacker

Worried mother

server

Record list: 1000

Attacks summary

	Attacker without traffic knowledge	**Attacker with traffic knowledge**
Server without authentication	Permissive definition	Not covered in the RFC
	call – id and cseq verification	SOHA solution (our solution)
Server with authentication	Impossible to attack	Permissive implementation
	------------------	Zero duration nonce value

Problem Solution

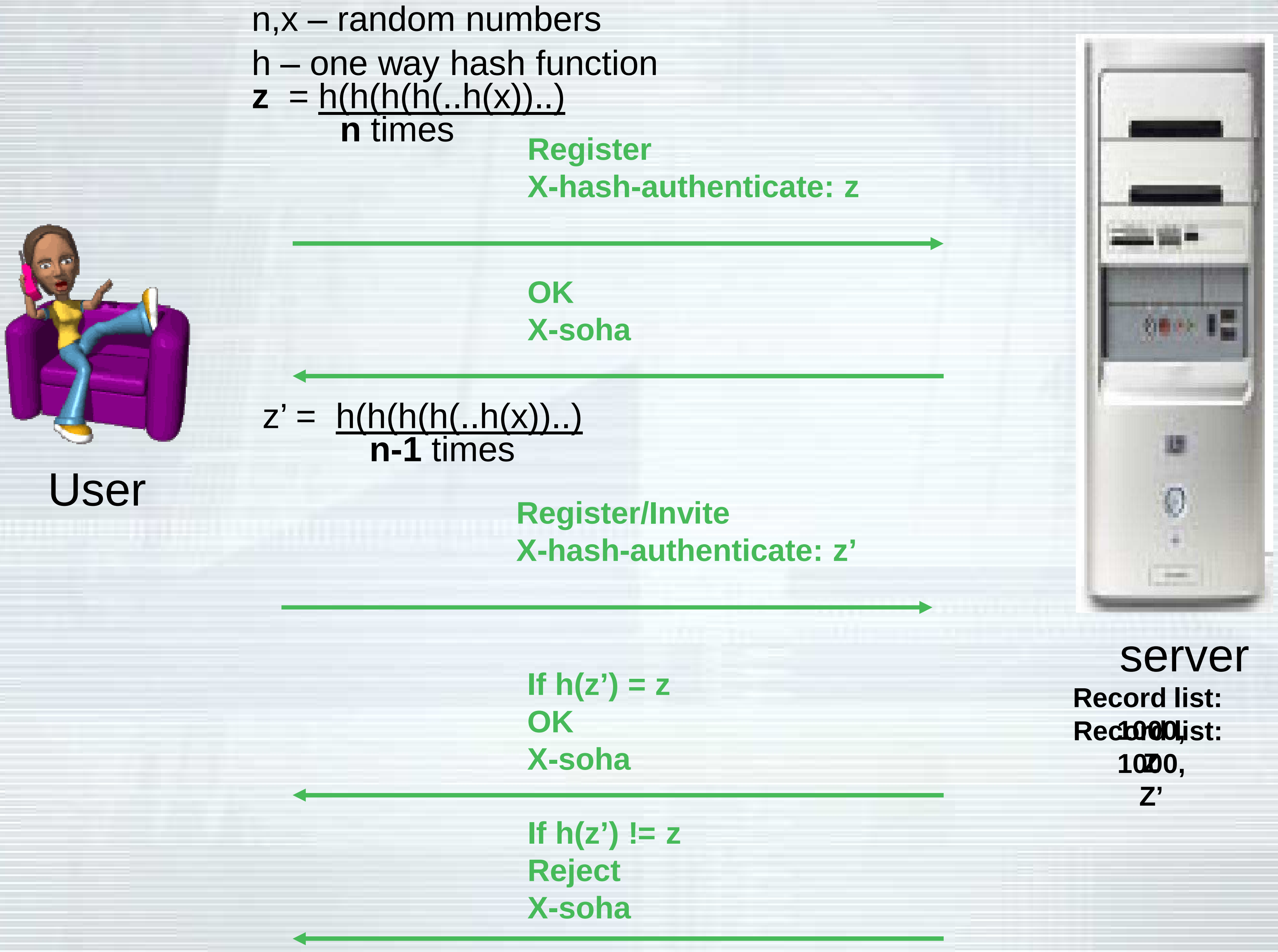
- Introduction to Telephony
- VoIP
- SIP
- Unregister Attack
- SOHA Solution
- Conclusion

Sip **O**ne **w**ay **H**ash function **A**lgorithm - SOHA

- Provides a protection from the attack of server without authentication and attacker with traffic knowledge
- Provides protection from all other attacks as well
- Does not require configuration changes
- Based on “**first is exclusive**” rule - the first user to capture the record becomes the exclusive user

SOHA (cont.)

- Hash function –
  - Takes a variable-length string as the input
  - Produces a fixed-length value as the output.
- One way function - a function that follows the following rules:
  - The description of the function is known and does not require any secret information for its operation.
  - H(x) => Y
  - Y => H(?)



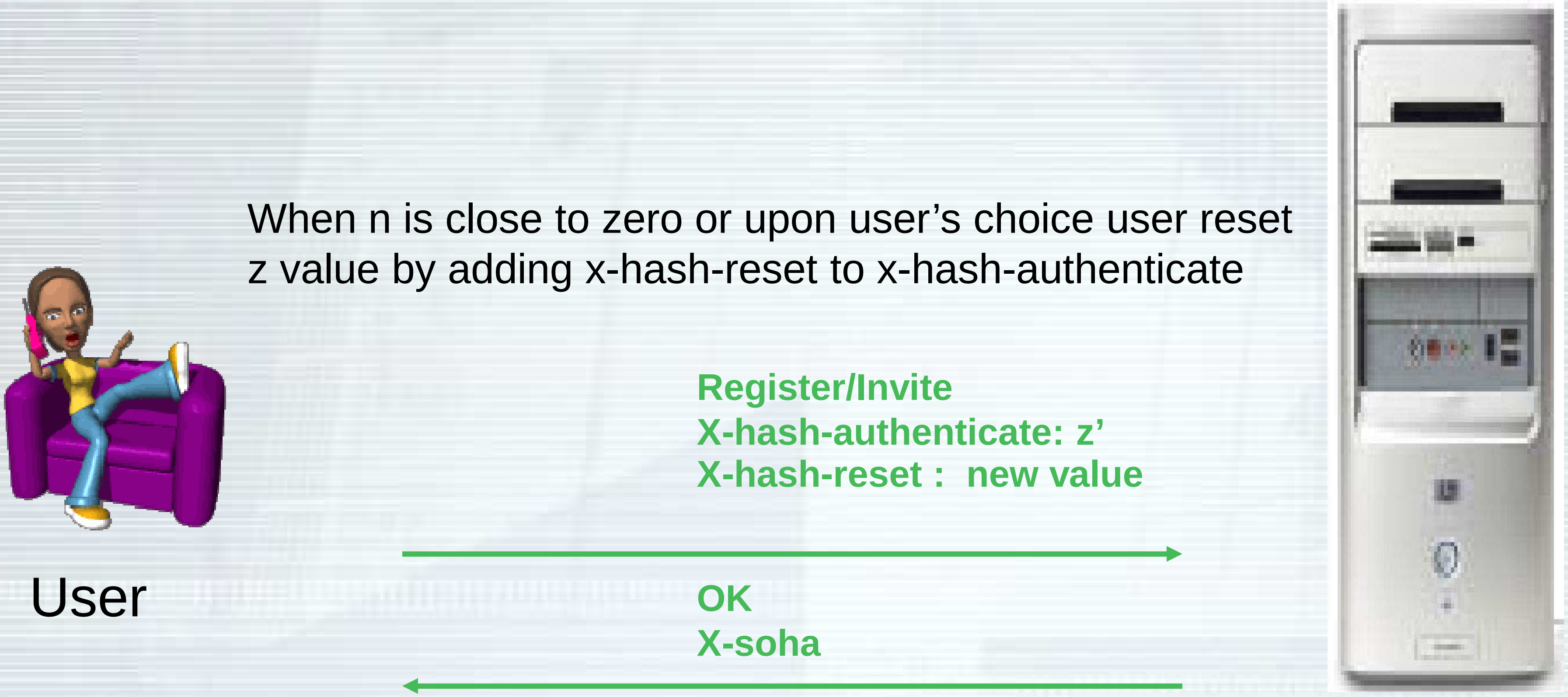
n,x – random numbers

h – one way hash function

$$\mathbf{z} = \underbrace{h(h(h(h(..h(x))..)}_{\mathbf{n}\text{ times}}$$

$$z' = \underbrace{h(h(h(h(..h(x))..)}_{\mathbf{n-1}\text{ times}}$$

When n is close to zero or upon user’s choice user reset z value by adding x-hash-reset to x-hash-authenticate



User

Register/Invite
X-hash-authenticate: z’
X-hash-reset : new value

OK
X-soha

server

Record list:
1000, new val

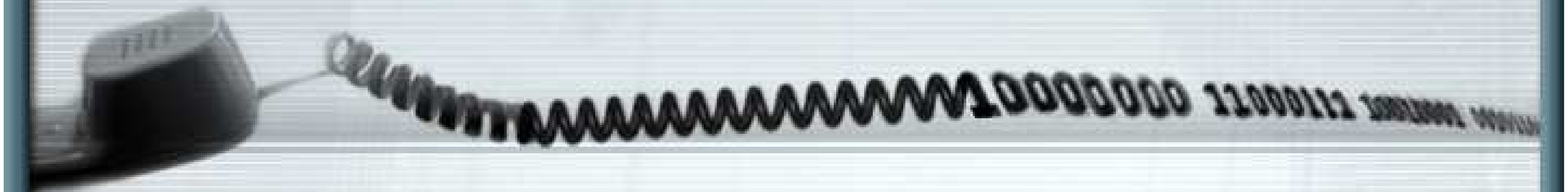
Record list:
1000, z

SOHA (cont.)

- Does not verify the identity of the user
- Ensures that a correctly registered user will not be removed from the server by the attacker

SOHA (cont.)

- Fully backward compatible
- Requires an addition of header fields (supported by SIP RFC):
  - *x-hash-authenticate* – used by the client.
  - *x-hash-reset* – used by the client.
  - *x-soha* – used by the server to indicate it supports SOHA
- SOHA similar to one time key password (RFC 2289)



- Introduction to Telephony
- VoIP
- SIP
- Unregister Attack
- SOHA Solution
- Conclusion

Conclusion

- With advancement of telephony comes a new set of possible attacks
- The attacks are either on VoIP protocols or on supplementary protocols (UDP,DNS etc)

Conclusion (cont.)

- Some of the attacks can be prevented with strict implementation of the RFC
- It is worth considering changing some of the RFC requirements from SHOULD to MUST to prevent possible attacks
- The consequence of implementing a SHOULD mechanism is not clear
  - non authentication in the server => unregister attack

Questions?