AMAZING

ENCRYPTION PROTOCOLS

S

M

M

M

TLS

SRTP

DTLS
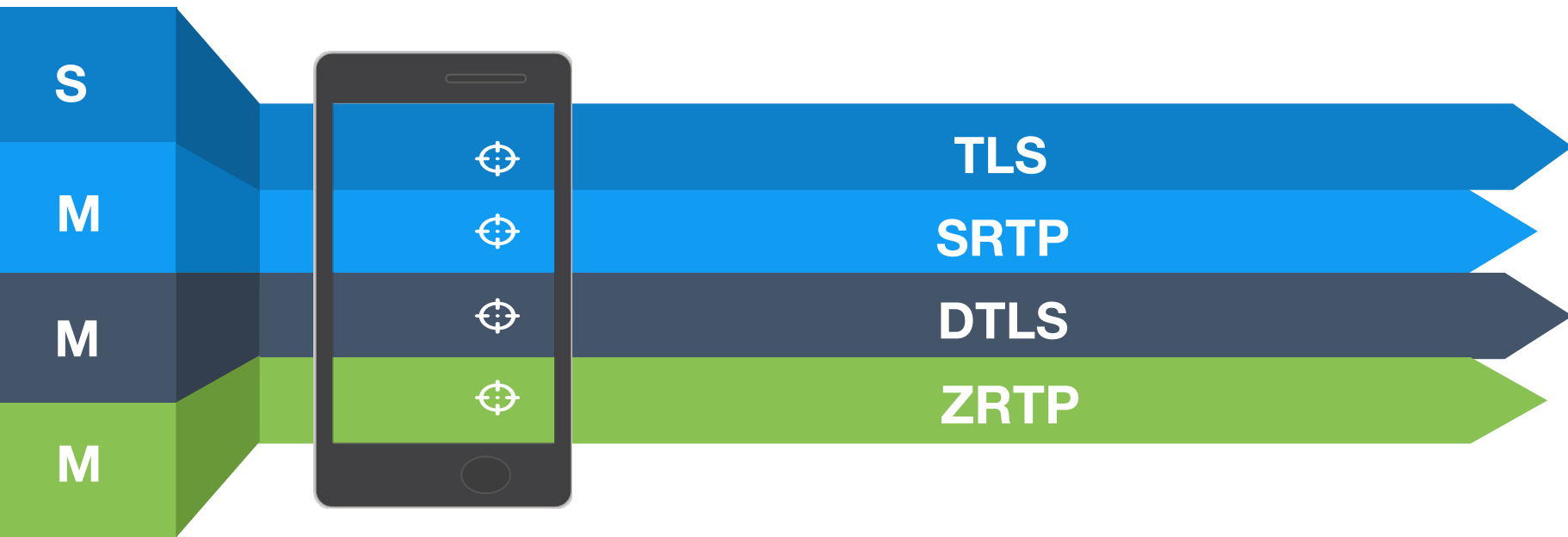
ZRTP

Secure Communications

What is ZRTP encryption for voice?

ZRTP, short from Zimmermann Real-time Transport Protocol, is a cryptographic key-agreement protocol meant to negotiate the keys for encryption between two end points in Voice-over-Internet-Protocol (VoIP) telephony.

In other words, ZRTP provides end-to-end encryption for VoIP calls. Not cellular calls or other types of telephony, only Voice over IP.
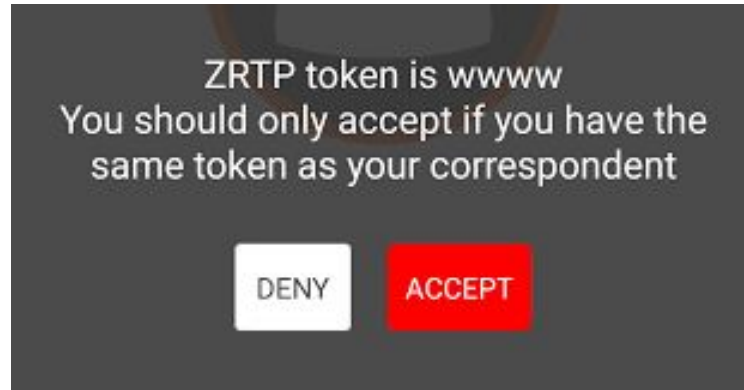
How Secure is ZRTP?

If you want a truly anonymous life, then maybe it's time you learned about Tor, CSpace and ZRTP.

These three technologies could help people <u>hide their activities</u> from the National Security Agency, according to <u>NSA documents</u> newly obtained from the archive of former contractor Edward Snowden by the German magazine <u>Der Spiegel</u>.

The combination of Tor, CSpace and ZRTP (plus another anonymizing technology for good measure) results in levels of protection that the NSA deems "catastrophic" — meaning the organization has "near-total loss/lack of insight to target communications,"!
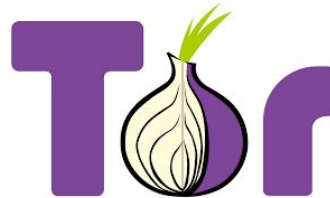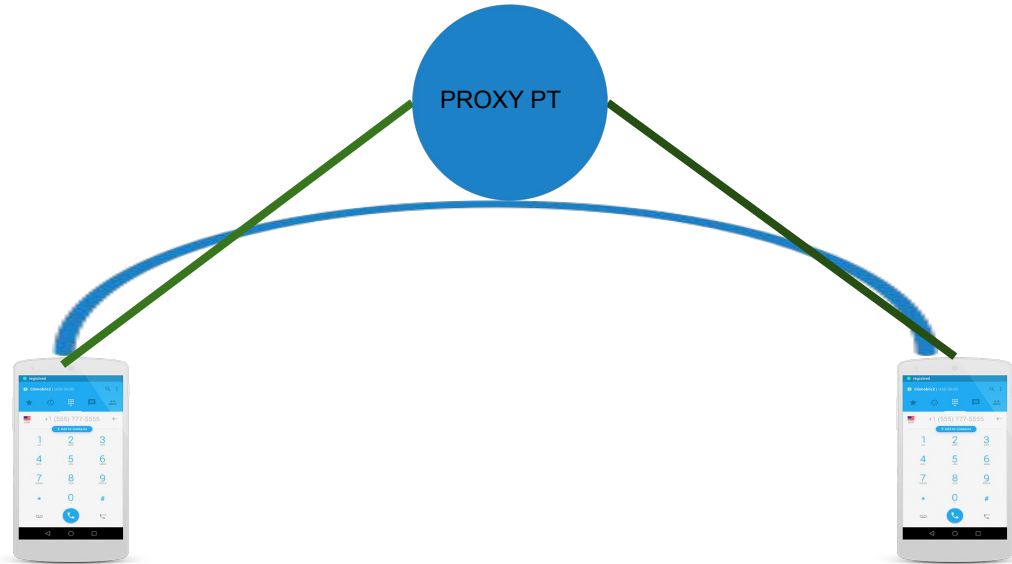
What's the magic?

- No need to support Peer to Peer or ICE
- Media must be unaltered

PROXY PT

MYSTERY CUSTOMER 2013

BEFORE 8bit

?

DURING 8bit

Secure Communications

ZRTP NATIVE SUPPORT

Cryptography

iOS	ANDROID
Security Protocols **TLS-1.2,-DTLS-1.2-SRTP-ZRTP-SDES-OTR**	

Symmetric ciphers

AES (128,192,256)

Modes of operation

CBC
CFB
CTR
AEAD-CCM
AEAD-GCM
XTS

Hash Function

MD5
SHA-1
SHA-224
SHA-256
SHA-384
SHA-512

Public Key algorithms

RSA
DSA
Delffie-Helmann (DH)
Elliptic Curve DSA
Elliptic Curve DH

Secure Random numbers (iOS) SecRandomCopyBytes

Secure random numbers (Android) /dev/urandom

GERMAN RESEARCHERS

Table 2. Evaluation results for the most common ZRTP-capable VoIP clients using our protocol and non-protocol tests (ascending alphabetical order by name).

Application	OS	Version	Library	Protocol Tests						Non-Protocol Tests				
				[zrtpCall]	[verDown]	[weakDH]	[invSS]	[invCom]	[sharedMitM]	[pbxEnroll]	[statusInd]	[confSAS]	[termError]	[secDef]
Acrobits Softphone	iOS	5.8.1	-	●	●	●	●	●	●	–	○	●	○	○
CSipSimple	Android	1.02.03	ZRTP4PJ	●	●	●	○	●	○	–	◑	●	○	●
Jitsi	Win, Lin, MacOS	2.9.0	ZRTP4J	◑	●	●	●	●	○	–	●	●	○	●
Linphone Android	Android	3.1.1	bzrtp	●	●	●	○	○[a]	○	–	○	●	○	○
Signal	Android	3.15.2	-	●	–	●	–[b]	●	–[b]	–	●	○	●	●
Signal	iOS	2.6.4	-	●	–	●	–[b]	●	–[b]	–	●	○	◑	●

● = pass, ◑ = partially, ○ = fail, – = not supported
[a] CVE-2016-6271
[b] Signal is a *cacheless implementation*. It does not support Preshared mode.

5.1 Acrobits Softphone

As shown in Figure 4, Acrobits Softphone behaved perfectly in all protocol tests. It is the only implementation that implemented the labeling of ZIDs and thus provides protection against [sharedMitM]. For [verDown],

http://www.ibr.cs.tu-bs.de/papers/schuermann-popets2017.pdf
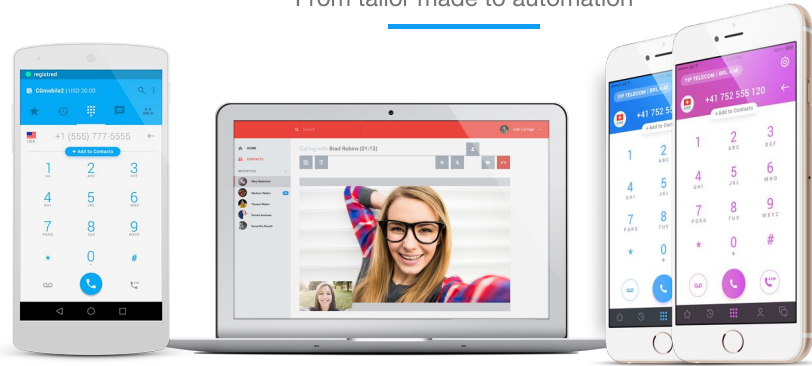
Most Secure "Class" 5 in the World

VAS or Sales Differentiator

Cloud Softphone 2.0

From tailor made to automation

• Acrobits's Cloud Softphone is a multi awarded cloud provisioning device portal, highly configurable to build and manage Mobile and WebRTC endpoints on the fly. Providers can build their projects without any programming knowledge. The feature set can be manipulated to create OTT application as well as complex Unified Communications services

Forwarding & Transfering
Attended And Un-Attended Transfers

Encryption
Military Grade Encryption Technology with ZRTP SRTP TLS

Conferencing
Audio or Video Conferencing with Call Recording

Contacts Synchronization
Contact Synchronisation Between Mobile and Desktop Apps

External Provisioning
"One-Touch" Provisioning and "Over-the-Air" updates

API
Powerful APIs for "under-the-hood" Customisations

SIP Server Agnostic
Connect to any SIP-IMS-RCS enable infrastructure

Do it Yourself
Design,Configure,Test and Deploy Like a Rocket

Messaging
Bi-Directional via SIP-SIMPLe or Web-Service API

Multiplatforms
Standalone APPs for Smartphones and Desktop webRTC Clients

• Acrobits's Cloud Softphone was designed to overcome the Operating systems fragmentation challenge which was over the years dramatically increasing the development and maintenance cost. Devices OSs, specially the Android one, come in all shapes and sizes, with vastly different performance levels and screen sizes. Furthermore, there are many different versions of Android that are concurrently active at any one time, adding another level of fragmentation. What this means is that developing apps that work across the whole range of Android devices can be extremely challenging and **time-consuming**