Confidenciabilidade e Integridade em Chamadas VoIP Com Troca de Chave Segura Utilizando o protocolo ZRTP

Abstract. *This paper describes the ZRTP protocol, a key agreement protocol, explaining its operations and how to guarantee a secure key agreement. These keys will be used by SRTP protocol that is responsible to encrypt and decrypt voice packets during a VoIP call.*

Resumo. *Este artigo descreve o protocolo de negociação de chaves ZRTP explicando sua forma de operação e de que forma ele garante a troca de chaves segura para que esta seja utilizada pelo protocolo SRTP para realizar a encriptação e decriptação dos pacotes de voz durante uma chamada VoIP.*

1. Introdução

Atualmente, muito se fala em formas de redução de custos com comunicação, neste cenário uma excelente alternativa é o VoIP que cada vez mais tem se tornado uma opção atrativa para as corporações, como também para usuários domésticos. O próprio governo Federal já vem utilizando o VoIP como meio de comunicação e como boa prática para diminuir seus custos neste segmento.

O desenvolvimento desta tecnologia tem tido como foco principal a redução de custos com comunicação, os projetos de implantação das redes VoIP geralmente não observam as questões de segurança, o que têm deixado toda a infra-estrutura de Tecnologia de Comunicação e Informação (TIC) vulnerável a ataques. Exemplos destes são o *eavesdropping*, muito semelhante a nossa tradicional escuta telefônica e a modificação do áudio [Dwivedi 2009]. Para exemplificar melhor o primeiro tipo de ataque imaginemos que um usuário A que aqui chamaremos de Alice, realiza uma determinada chamada para outro usuário B, identificado aqui como Bob tendo neste canal um intruso I conhecido por Eve. O intruso I utilizando-se de técnicas de *sniffing* dos pacotes, que trafegam no canal onde se comunicam os usuários A e B, é capaz de reconstruir um determinado ponto do fluxo de áudio e conseguir até reproduzir a chamada VoIP entre Alice e Bob quebrando assim a confidenciabilidade na comunicação. Exemplificando melhor o segundo ataque, Eve pode ser capaz de inserir ou mixar mensagens gravadas previamente dentro da conversa ativa entre Alice e Bob, desta forma eles poderão ouvir informações que não foram faladas nem por Alice e nem por Bob, quebrando assim a integridade da comunicação entre eles [Endler e Collier 2007].

Já existem alguns países da comunidade européia que criaram uma legislação que permite a monitoração de todo o tráfego que entra e sai de suas fronteiras, isso inclui o tráfego de pacotes VoIP [Olsen 2009]. Mediante fatos como este, a utilização e suporte a ferramentas que garantam a confidenciabilidade e integridade das chamadas VoIP se torna algo de grande relevância para usuários e provedores de serviços VoIP.

Neste artigo será apresentado o ZRTP que é um protocolo de negociação de chaves criptográficas, que serão utilizadas para encriptação da mídia (voz). Este protocolo procura garantir a confidenciabilidade e a integridade entre as chamadas telefônicas entre os usuários Alice e Bob impedindo desta forma que I utilize técnicas

como as supra citadas . Como o ZRTP ainda é um *internet*-d*raft*[1] [Zimmermann, Johnston e Callas 2009], ainda não existem muitas implementações de *endpoints* que suportem este protocolo, no entanto, será utilizado o o S*oftphone Twinkle* que é licenciado sobre GPL(*General Public License*) e permite a utilização do ZRTP, conforme será detalhado no tópico 5.

2. VoIP: Fundamentos

Para utilização da tecnologia VoIP, se faz necessário alguns elementos fundamentais desta arquitetura:

- Servidor *Proxy*: É o responsável por manter o cadastro dos usuários, bem como sua localização. É ele quem processa as chamadas e realiza o encaminhamento das mesmas entre os *endpoints*. Para o estudo de caso apresentado no tópico 5 será utilizado o software *Asterisk* como servidor Proxy.
- *Endpoints*: São os equipamentos ou softwares que representam os usuários. São neles onde são configuradas as extensões ou números ou ainda ramais dos usuários. Eles são a interface utilizada pelo usuário para realização da chamada telefônica. Para o estudo de caso apresentado no tópico 5 será utilizado o *softphone Twinkle*.

Para que uma chamada seja completada são necessários dois tipos de protocolos: Protocolos de sinalização e protocolos de mídia. Protocolos de sinalização são responsáveis pelo estabelecimento e finalização de chamadas e negociação de mídia. Já protocolos de mídia, são responsáveis por transportar a voz propriamente dita e supervisionar a qualidade da chamada. O protocolo SIP (*Session Initiation Protocol*), descrito pela RFC 3261 [Davidson, Peters, Bhatia, Kalidindi e Mukherjee 2008], é o mais utilizado para sinalização da chamada, o protocolo SDP (*Session Description Protocol*), descrito pela RFC 4566 [Davidson, Peters, Bhatia, Kalidindi e Mukherjee 2008], para a e negociação de mídia e o protocolo RTP (*Real-time Transport Protocol*), descrito pela RFC 3550 [Davidson, Peters, Bhatia, Kalidindi e Mukherjee 2008], para o transporte da mídia.

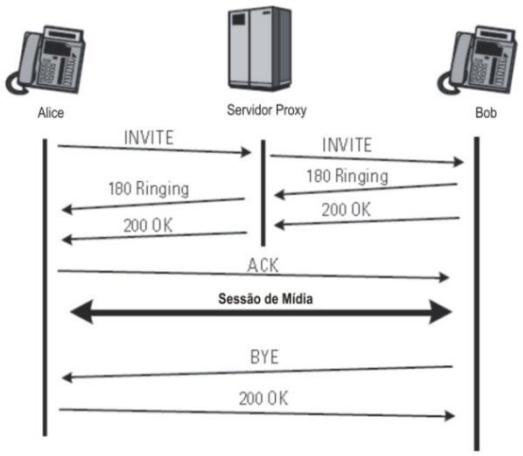

Figura 1. Mensagens de uma chamada VoIP

[1] São documentos que estão sendo trabalhados pelo *Internet Engineering Task Force* (IETF), e suas áreas e grupos de trabalho.

Protocolos de sinalização são utilizados na comunicação entre os *endpoints* e o *Proxy*, uma vez que a sinalização seja finalizada, dá-se inicio a conversação, neste ponto a mídia (voz) transportada pelo protocolo RTP, pode ser enviada para o *proxy* por Alice e em seguida o *proxy* encaminha para Bob. Esta ação é chamada de *relay* de mídia. m Em outro cenário possível, representado na Figura 1, Alice e Bob trocam pacotes RTP diretamente sem a necessidade de intermediação do servidor *proxy*. Este será o cenário utilizado para o estudo de caso apresentado no tópico 5.

2.1. Encriptação de Mídia

Conforme descrito na introdução, durante a troca de pacotes RTP entre Alice e Bob o atacante aqui representado por Eve pode ser capaz de empreender um ataque do tipo *eavesdropping* onde ao final do mesmo, Eve terá um arquivo de áudio contendo toda a conversa entre Alice e Bob ou ainda um ataque de modificação de áudio. Neste cenário, se faz o uso do protocolo SRTP (*Secure Real-time Transport Protocol*), que é uma extensão do protocolo RTP que fornece funcionalidades de segurança ao RTP como por exemplo encriptação e autenticação. Ele é descrito pela RFC 3711 [Park 2009]. Durante a negociação da chamada, os *endpoints* trocam as chaves e em seguida começam a encriptar/decriptar os pacotes RTP [Park 2009].

O SRTP utiliza um processo de derivação de chave para execução de sua tarefa. Ele recebe do protocolo de gerenciamento de chaves a chave mestra acrescida de um sal[2] e a partir desta, são derivadas seis chaves de sessão que são válidas durante um contexto específico e são constantemente renovadas e descartadas.

A RFC 3711 não descreve como o protocolo de gerenciamento de chaves deve fazer para enviar a chave mestra para Alice e Bob [Park 2009], podendo ser implementado de diversas formas.

No próximo tópico será descrito o protocolo ZRTP que é utilizado justamente para realizar a tarefa de envio da chave mestra para Alice e Bob de forma segura.

3. Protocolo ZRTP

O ZRTP é um protocolo de negociação de chaves, que realiza a troca de chaves utilizando o esquema Diffie-Hellman durante a negociação da chamada, ele faz uso da mesma porta utilizada pelo fluxo RTP e não necessita de suporte do protocolo de sinalização [Zimmermann, Johnston e Callas 2009]. O protocolo ZRTP também não requer uma Infra Estrutura de Chaves Públicas (PKI)[3] ou uso de certificados pelos *endpoints*. Para a sessão de mídia, o ZRTP garante a confidencialidade e integridade, protegendo a sessão contra ataques de *man-in-the-middle (MiTM)*, e autenticação em casos onde o protocolo de sinalização provê proteção de integridade fim-a-fim. O ZRTP pode utilizar os atributos do Protocolo de Descrição de Sessão (SDP), para prover a capacidade de verificação do suporte ao ZRTP através do canal de sinalização. Uma vez validado o suporte ao ZRTP pelos *endpoints* da chamada é gerado um segredo compartilhado que é utilizado na geração de chaves e sal para o SRTP.

---

[2] Um sal é utilizado para dar uma maior complexidade a chave. Através da adição do sal, ataques de força bruta contra o texto encriptado, tornam-se praticamente impossíveis de serem realizados [Park 2009].

[3] É um conjunto de hardware, software, pessoas, políticas, e procedimentos necessários para criar, gerenciar, armazenar, distribuir e revogar certificados digitais [Burnett e Paine 2002].

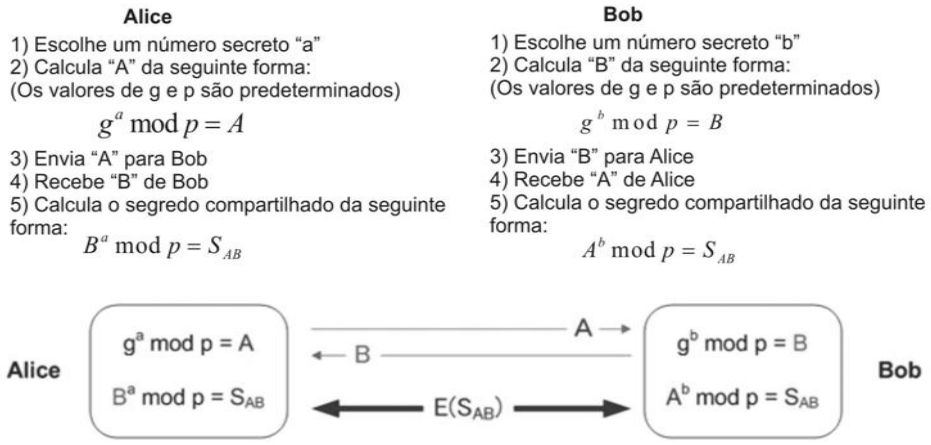

Figura 2. Geração do segredo compartilhado [Chen 2006]

Uma das funções criptográficas utilizadas pelo ZRTP que é um diferencial em relação a outros protocolos semelhantes é a utilização de *Diffie-Hellman* (DH) com comprometimento de hash permitindo a detecção de ataques de *man-in-the-middle* (MiTM) através da exibição de um *short authentication string* (SAS) para os usuários. Durante a chamada, Alice solicita verbalmente a Bob o código SAS que foi exibido para ele na interface do endpoint(Figura 4) se o código exibido para Alice foi igual ao que foi exibido para Bob, significa que a chamada está segura contra o atacante Eve. As chaves utilizadas são destruídas ao final da chamada, o que evita o comprometimento da chamada em função de divulgação das informações da chave daquela sessão [Zimmermann, Johnston e Callas 2009].

Tudo isso é feito sem dependência de uma infra-estrutura de PKI, certificação de chave, modelos de confiança, entidades certificadoras ou qualquer outro gerenciamento complexo de chaves que tanto atormentam os administradores de segurança de TI . O ZRTP também não confia na sinalização SIP para gerenciamento de chaves, e de fato não confia em nenhum servidor, por isso, realiza a negociação e o gerenciamento de chaves puramente *peer-to-peer*[5] sobre o *stream* RTP.

O protocolo ZRTP pode ser usado e detectado sem que seja declarado na negociação da sinalização. Desta forma, também se reduz a complexidade de sua implementação e minimiza a interdependência entre as camadas de sinalização e de mídia. No entanto, quando o ZRTP é indicado na sinalização através do atributo SDP *zrtp-hash*, são habilitadas funcionalidades bastante úteis. Enviando na camada de sinalização um *hash* da mensagem ZRTP *Hello*, o ZRTP provê uma ligação entre as camadas de sinalização e de mídia. Quando isso é feito através de um protocolo de sinalização que possui proteção de integridade fim-a-fim, a troca do DH é automaticamente protegida contra ataques de *MiTM* [Zimmermann, Johnston e Callas 2009].

[5] Toda a negociação da chave é feita entre o usuário A e o usuário B sem o envolvimento ou interferência de nenhum outro elemento.

O ZRTP é negociado da mesma forma como uma sessão RTP é negociada O protocolo ZRTP começa depois que dois *endpoints* tenham utilizado um protocolo de sinalização como o SIP e estão prontos para transmitir a mídia.

Para casos em que o uso do protocolo de sinalização para detecção do ZRTP não seja possível, o ZRTP pode incluir uma marcação em um pacote normal RTP logo no inicio da sessão de forma que o outro endpoint possa detectar o se haverá ou não suporte ao ZRTP. Se o outro endpoint for capaz de ler esta marcação, então ambos já sabem que suportam o ZRTP.

A validação do suporte ao protocolo ZRTP é feita automaticamente pelos *endpoints*. Um *endpoint* que possua suporte ao protocolo ZRTP envia uma mensagem ZRTP *Hello* para o outro *endpoint, onde* o objetivo desta mensagem é confirmar se o outro *endpoint* suporta o protocolo e também para verificar quais algoritmos os dois *endpoints* possuem em comum (Figura 3).

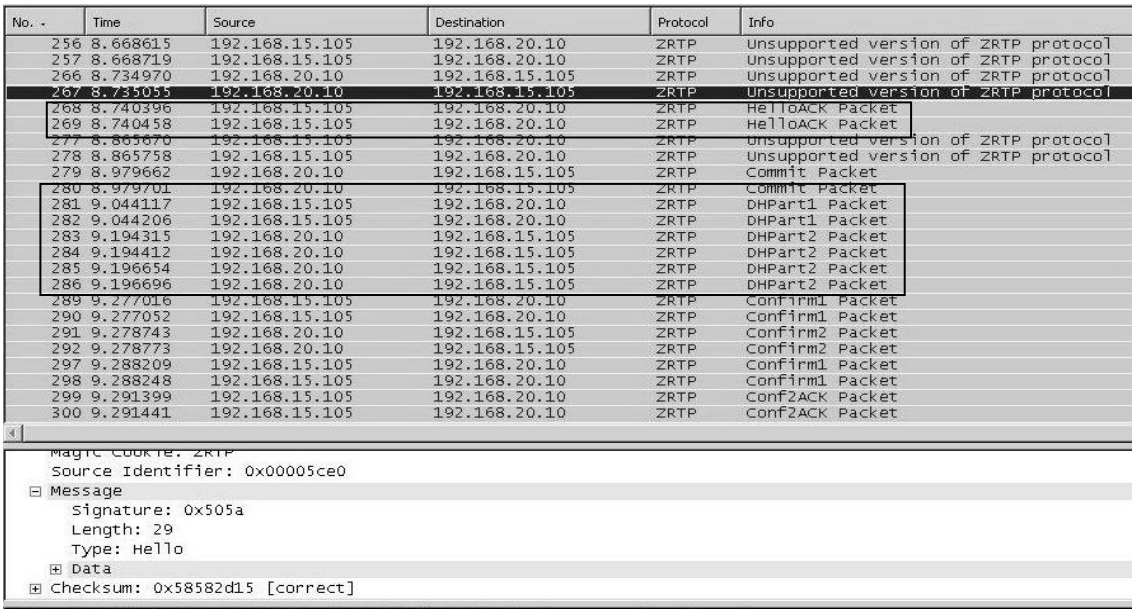

Figura 3. Negociação ZRTP

A mensagem *Hello* contém as opções de configuração para ser utilizada na abertura da sessão SRTP. Cada instancia do ZRTP possui um único e randômico ZRTP ID ou ZID de 96-bits. O ZID também compõe a mensagem *Hello*. O ZID recebido através das mensagens *Hello* é utilizado para localizar segredos compartilhados utilizados em sessões anteriores com aquele *endpoint*(dententor do ZID em questão) [Bresciani 2007].

A resposta a uma mensagem ZRTP *Hello* é uma mensagem do tipo ZRTP *HelloACK* (Figura 3), *e*sta mensagem simplesmente confirma a recepção da mensagem de *Hello*. A mensagem de *Hello* e outras mensagens ZRTP possuem um *hash* que é utilizado para ligar as mensagens. Isso permite a rejeição de falsas mensagens ZRTP durante uma negociação.

Uma vez recebido o *HelloACK*, ambos *endpoints* já sabem os parâmetros que devem ser utilizados durante a sessão SRTP e que o ZRTP é suportado para realização da negociação da chave a ser utilizada pelo SRTP.

4. Twinkle

O *Twinkle* é um *softphone* para ser utilizado com comunicação VoIP e mensagens instantâneas, fazendo uso do o protocolo de sinalização SIP. Ele está disponível apenas para o sistema operacional Linux e é licenciado sobre a GPL [Boer 2007].

Uma grande vantagem deste *softphone* é que ele já vem com suporte ao ZRTP/SRTP nativo, permitindo a realização de chamadas VoIP seguras com outros clientes que também suportem o ZRTP.

No momento da escrita deste trabalho sua versão era a 1.4.2 e ele implementa a versão 0.85 do protocolo ZRTP.

5. Estudo de Caso

Para demonstração dos conceitos aqui apresentados, foi montado um laboratório, usando ferramentas *open source* para implementar e analisar os protocolos e cenários aqui discutidos. Como servidor proxy, foi utilizado um servidor *asterisk* na versão 1.6.0.2, rodando o sistema operacional Linux Centos 5.3 32bits, dois *notebooks* rodando o sistema operacional Linux Fedora 10 de 64 bits com o *Twinkle* 1.4.2 instalado em ambos e para captura dos pacotes foi utilizado o software *wireshark*.

Como o ZRTP roda muito próximo da placa de rede, não é possível capturar os pacotes encriptados na mesma máquina em que está o *softphone* [Zimmermann 2006], por isso o tráfego entre os dois clientes foi roteado através do servidor (não é *relay* e sim roteamento), de forma que as capturas de pacotes pudessem ser feitas neste servidor ao invés de ser feita nos *endpoints*.

Como o suporte ao ZRTP será garantido pelos *endpoints* é fundamental que o *asterisk* não faça *relay* de mídia, caso contrário quando os *endpoints* enviarem a mensagem de ZRTP HELLO, o *asterisk* não irá respondê-la, uma vez que o mesmo não está instalado com suporte ao protocolo ZRTP. Para tanto, nas configurações dos *endpoints*, ou seja, dentro do arquivo de configuração /etc/asterisk/sip.conf[8], devemos adicionar o parâmetro canreinvite=yes.

[4101]	[4102]
type=friend	type=friend
callerid=R4101<4101>	callerid=R4102<4102>
secret=4101	secret=4102
host=dynamic	host=dynamic
canreinvite=yes	**canreinvite=yes**
nat=no	nat=no
context=default	context=default

Quadro 1: Configurações das extensões utilizadas

Após a configuração do *Twinkle* nos dois clientes com a extensão[9] 4101 e outro com a 4102, realizamos a chamada inicial. Na interface do *Twinkle* é apresentado um

[8] Arquivo utilizado para configurar os parâmetros do protocolo SIP do servidor asterisk

[9] Ramal

cadeado e um código, estes confirmam que a chamada está sendo encriptada. O código que é exibido é o SAS e permite que as duas partes se autentiquem verbalmente, pois o código exibido para os dois clientes tem que ser o mesmo [Zimmermann e Johnston e Callas 2009]. Esta chamada está sendo realizada de forma segura, encriptando os pacotes RTP através do SRTP.

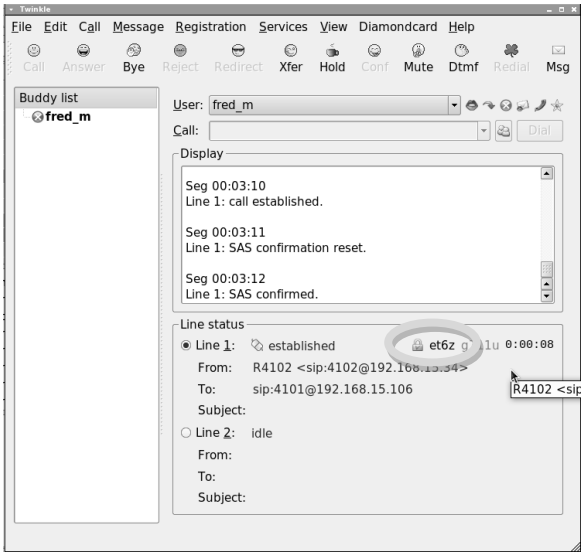

Figura 4. Chamada ativa no *Twinkle* com indicação do ZRTP e SAS

6. Considerações Finais

O protocolo ZRTP nos fornece uma excelente opção para tornar segura as comunicações VoIP, no entanto, ainda não foi aprovado pelo IETF[11] e ainda não virou um RFC e por isso não é suportado pelos principais endpoints do mercado. No entanto, a comunidade de software livre inovou mais uma vez adicionando suporte ao ZRTP no *softphone Twinkle*, permitindo que seus usuários estabeleçam conexões seguras através da utilização do protocolo ZRTP. Nos testes executados, mesmo tendo a voz encriptada, não foi percebida degradação da qualidade do áudio, nem *delay* na inicialização da chamada.

Mesmo usando a solução apresentada, ainda existem cenários de chamada em que o cliente não estará seguro contra ataques do tipo *eavesdropping*, como por exemplo: quando são realizadas chamadas para a rede PSTN (*public switched telephone network*), pois o *endpoint* final não suporta o ZRTP , por isso, a máxima segurança que se consegue é até o *media gateway* que entregará esta chamada a PSTN.

Outro cenário onde a negociação de chaves não irá funcionar é quando o *proxy* realiza *relay* da mídia, caso muito comum nas implementações comerciais, desta forma, ao receber o pacote SDP contendo em seu cabeçalho o campo *zrtp-hash* vindo do cliente, o *proxy* irá reescrever este pacote SDP e reenviará para o destino só que como o *proxy* não possui suporte ao protocolo ZRTP, ele não irá incluir o header *zrtp-hash* indicando que o protocolo ZRTP não é suportado por ele.

[11] Internet Engineering Task Force

Desta forma, fica como recomendação de estudos futuros, a implementação do protocolo ZRTP no *proxy* da rede de forma a garantir que ele seja capaz de suportar o ZRTP mesmo sendo feito *relay* de mídia, outro ponto levantado é a interoperabilidade entre versões do ZRTP, os *endpoints* com versões mais atuais devem ser capazes de se comunicar com os *endpoints* de verão mais antigas, fato este que não se comprovou em nosso estudo de caso.

7. Referências

Park, P. (2009). Voice over IP Security, páginas 71,188-193. Cisco Press.

Dwivedi, H. (2009). Hacking VoIP: protocols, attacks, and countermeasures, páginas 179-188. No Starch Press.

Endler, D and Collier, M. (2007). Hacking Exposed VoIP: Voice over IP Security Secretes and Solutions, páginas 471-481. McGraw Hill.

Burnett, S., Paine, S. (2002). Criptografia e Segurança. O Guia Oficial RSA, páginas 89-93,145-177. RSA Press.

Davidson, J., Peters, J., Bhatia, M., Kalidindi, S., Mukherjee, S. (2008). Fundamentos de VoIP, páginas 271-301. Bookman.

Zimmermann, P., Johnston, A., and Callas, J.. (2009) "ZRTP: Media Path Key Agreement for Secure RTP", IETF Draft; http://www.ietf.org/internet-drafts/draft-zimmermann-avt-zrtp-15.txt

Bresciani, Riccardo (2007) "The ZRTP Protocol: Security Considerations", http://uni.nopkoguo.net/pub/ZRTP_Protocol-Security_Considerations-PaperNSS07.pdf, Março.

Olsen, Ruben. (2009) "European legislation will force usage of encrypted VoIP".

http://voipsa.org/blog/2009/04/11/european-legislation-will-force-encrypted-voip/. Abril.

Chen, Eric. (2006). "A Tour Through Zfone". http://voipsa.org/blog/2006/06/19/a-tour-through-zfone , Março

Zimmermann, P. (2006) "Wireshark (Ethereal) ZRTP packet dissector". http://www.zfoneproject.com/wireshark.html, Abril.

Boer, M. (2007) "Twinkle". http://www.twinklephone.com/, Fevereiro.