https://www.emeraldinsight.com/doi/full/10.1108/ICS-07-2014-0046

Privacy-preserving, User-centric VoIP CAPTCHA Challenges: an Integrated Solution in the SIP Environment

Aimilia Tasidou, Pavlos Efraimidis, Yiannis Soupionis, Lilian Mitrou, Vasilis Katos

Abstract

Purpose - In this work we argue that it is possible to address discrimination issues that naturally arise in contemporary audio CAPTCHA challenges and potentially enhance the effectiveness of audio CAPTCHA systems by adapting the challenges to the user characteristics.
Design/methodology/approach - We design a prototype, called PrivCAPTCHA, to offer privacy-preserving, user-centric CAPTCHA challenges. Anonymous credential proofs are integrated into the SIP protocol and the approach is evaluated in a real-world VoIP environment.
Findings - The results of this work indicate that it is possible to create VoIP CAPTCHA services offering privacy-preserving, user-centric challenges, while maintaining sufficient efficiency.
Research limitations/implications - The proposed approach was evaluated through an experimental implementation to demonstrate its feasibility. Additional features, such as appropriate user interfaces and efficiency optimizations, would be useful for a commercial product. Security measures to protect the system from attacks against the SIP protocol would be useful to counteract the effects of the introduced overhead. Future research could investigate the use of this approach on non-audio CAPTCHA services.
Practical implications - PrivCAPTCHA is expected to achieve fairer, non-discriminating CAPTCHA services, while protecting the user's privacy. Adoption success relies upon the general need for employment of privacy-preserving practices in electronic interactions.
Social implications - This approach is expected to enhance the quality of life of users, who will now receive CAPTCHA challenges closer to their characteristics. This applies especially to users with disabilities. Additionally, as a privacy-preserving service, this approach is expected to increase trust during the use of services that employ it.
Originality/value - To our knowledge this is the first comprehensive proposal for privacy-preserving CAPTCHA challenge adaptation. The proposed system aims at providing an improved CAPTCHA service that is more appropriate for and trusted by human users.

Keywords: Privacy Enhancing Technologies, Audio CAPTCHA, SPIT, Anonymous Credentials, Incident Response, Legal Aspects of Privacy, Discrimination, Jitsi, Idemix

Article Classification: Research paper

1. Introduction

The Internet Telephony (Voice over IP) is a developing technology that promises a low-cost and high-quality and availability service of multimedia data transmission. Inevitably though, VoIP "inherited" not only these positive features of Internet services, but also some obvious disadvantages (Walsh and Kuhn, 2005). One of the main disadvantages is Spam over Internet Telephony (SPIT) (El Sawda and Urien, 2006), which is the popular expression of Spam in VoIP network environments.

A CAPTCHA (Ahn *et al.*, 2004) is a method that is widely used to counter automated SPAM attacks. The same technique can be used to mitigate SPIT. A CAPTCHA is a Reverse Turing Test where a machine tries to identify whether the incoming session is initiated by a software application or a human. The three major categories of CAPTCHA are a) visual CAPTCHA, where the user tries to recognise characters or words in malformed pictures, b) audio CAPTCHA, where the characters or words to be recognised are in an audio file, and c) logic CAPTCHA, where the user tries to answer specific questions. This paper focuses on audio CAPTCHA. Visual CAPTCHAs are hard to apply in VoIP systems, mainly due to the limitations of end-user devices. Logic CAPTCHAs are well suited for the VoIP context and are appropriate for adaptive challenges, therefore we believe that the PrivCAPTCHA approach can also be applied to logic CAPTCHAs.

The audio CAPTCHA challenges used today to prevent automated SPIT attacks do not take into account the characteristics of the caller or the callee. The fact that these challenges are generic, does not allow for the process to take into consideration the cognitive abilities of human users, while at the same time discriminates against users that have difficulties solving the generic challenges. The ability to use information about the caller or the callee opens up new opportunities for creating more effective and fair CAPTCHA challenges. However, the required information about the caller will probably be sensitive personal information, and thus it is important that a privacy-preserving method of achieving the adaptation of CAPTCHA challenges is used.

This work proposes a user-centric, privacy-preserving VoIP CAPTCHA adaptation mechanism, offering personalised, non-discriminating challenges, while aiming at increasing the CAPTCHA mechanism effectiveness. The concepts and motivation of this work are discussed in Section 2. The proposed approach, including the utilised cryptographic building blocks, is described in Section 3. The implementation and tests created to integrate PrivCAPTCHA into the SIP environment are presented in Section 4. The conclusions of this work are discussed in Section 5.

1.1. Related work

CAPTCHA challenge solving is a highly user-dependent process. Works on the subject illustrate that the user characteristics can greatly affect the success rate of the

mechanism usage. As described in the following paragraphs, the user's sensory and cognitive abilities, computer literacy and language fluency play a decisive role on the user's ability to solve CAPTCHA challenges. In this section, we briefly survey the audio CAPTCHA technology, which is in the early stages of development. We focus first on evaluating the existing audio CAPTCHAs and then on recording the CAPTCHA attributes which improve its usability.

1.1.1. Existing CAPTCHA evaluation

Existing audio CAPTCHAs have been proven more difficult to use for visually impaired than non-visually impaired people (Bigham and Cavender, 2009). For their research they used 162 persons, of whom 89 were visually impaired, and popular website audio CAPTCHA implementations. Their research illustrated that audio CAPTCHAs are difficult to solve. Only 43% of users with visual impairments were able to answer an audio CAPTCHA at the first attempt and only 39% of other users. Moreover, it should be noted that visually impaired users took at least twice as long. Yet nearly half of the users (47%) still failed to respond correctly to an audio CAPTCHA after 3 attempts. This is a somewhat unexpected result, since one would anticipate that audio CAPTCHA challenges would be more appropriate for visually impaired persons.

Bursztein *et al.* (2010) conducted an extensive study on the ability of people to solve existing CAPTCHAs, as well. Regarding audio CAPTCHAs, they studied eight of the most popular implementations. The conclusions that emerged from their study were a) the period for listening and solving a CAPTCHA is certainly excessive (averaged over 25 seconds), b) the percentage of users who took second or third attempts, because the previous attempt was wrong, exceeded 50%, and c) people who were not native English speakers had major problems in solving the CAPTCHA and therefore the success rate was reduced by more than 20%.

Soupionis and Gritzalis (2010) classified the audio CAPTCHA attributes, evaluated the current popular audio CAPTCHA implementations and developed a new audio CAPTCHA for VoIP environments. The CAPTCHAs were classified based on their attributes into four categories: (a) vocabulary, (b) background noise, (c) time, and (d) audio production. Afterwards, the evaluation took place where the CAPTCHAs were utilised on the above mentioned attributes. The evaluation process was based on the fact that CAPTCHAs must be easy for human users to solve, easy for a tester machine to generate and grade, and hard for a software bot to solve. Therefore, the final evaluation was made by two means; namely, by user tests (~60 persons) and by two bots configured to solve audio CAPTCHAs. The evaluation process proved that a) the current CAPTCHA implementations are not adequate, meaning that every implementation is either too easy or too difficult to be solved by both users and bots, and b) the implementation attributes of some CAPTCHAs, like long vocabulary (> 8 characters) and language requirements (native vs. non-native English speakers), negatively affects the users' success rate (~40%) in most cases.

1.1.2. CAPTCHA usability

Yan and El Ahmad (2008) discuss usability issues that should be considered and addressed in the design of CAPTCHAs. The authors analyze a few aspects for the three main types of CAPTCHAs: 1) Text-based, 2) Sound-based, and 3) Image/picture-based schemes. As far as the Sound-based ones are concerned, the major parameters contributing to having a usable CAPTCHA are the following:

1) Distortion, meaning the background noise which distorts sounds in audio CAPTCHAs.

2) Content, meaning the content materials used in audio CAPTCHAs which are typically language specific, like digits, character set and string length.

3) Presentation, meaning the integration technique within the web pages which is still a great concern.

Lazar *et al.* (2012) describe the development of a new audio CAPTCHA called the SoundsRight CAPTCHA focusing on blind users. The authors identify that one of the major problems is the "linear audio CAPTCHA playback" problem. A blind user has to quickly navigate through a page using only keyboard strokes, and the screen reader causes audio interference with the audio CAPTCHA that is being played. Once the audio CAPTCHA is played, the user must quickly focus on the answer portion of the CAPTCHA to input what they heard. This problem may affect mainly the web implementation of audio CAPTCHA, but it shows that there should be a certain method for the CAPTCHA playback.

The above results indicate that there is a need and the potential to create more appropriate challenges for the human user that will allow for fewer problems in solving CAPTCHA challenges. Additionally, user-centric challenges provide the potential to weaken the connection between the difficulty of CAPTCHA solving for humans and for bots respectively, resulting in more effective CAPTCHA systems.

Taking into account the person's characteristics during CAPTCHA generation, brings about privacy concerns that need to be addressed. There has been significant progress on the subject of accountable, privacy-preserving services during the past decade. The privacy-preserving techniques used in the proposed system are closely related to accountable anonymous communication systems (Diaz and Preneel, 2007), anonymous credential systems (Camenisch and Pfitzmann, 2007) and electronic identity cards (Poller *et al.*, 2012, Deswarte and Gambs, 2010). Using cryptographic tools, all these systems aim at providing their functionality while protecting users' privacy. Similarly, we utilise existing cryptographic primitives to create a privacy-preserving personalised CAPTCHA system, which allows users to prove attributes about themselves and receive personalised challenges, without revealing their identity.

https://www.emeraldinsight.com/doi/full/10.1108/ICS-07-2014-0046

2. Concepts and Motivation

2.1. Problem statement and solution overview

The selection of an appropriate CAPTCHA challenge that successfully distinguishes between human users and bots is a challenging task. Generic CAPTCHA challenges that are difficult enough for bots, often pose difficulties to human users as well. Therefore, a method is needed to tailor CAPTCHA challenges closer to the human user, without necessarily lowering the difficulty level for bots. Overall, this work does not aim at making CAPTCHA challenges generally easier, rather, it aims at proposing a method to create more appropriate, effective and fair challenges for the users. However, the process of selecting the appropriate kinds of adaptations according to the user's characteristics is outside the scope of this work. We propose a privacy-preserving method of proving user characteristics to the CAPTCHA service and delivering the adapted challenge. We demonstrate the feasibility of the proposed system through representative examples.

One could argue that proving user characteristics to the VoIP service eliminates the need for the CAPTCHA test overall. However, we believe that the CAPTCHA test is still needed to protect users from unauthorised use of their accounts (hijacking) and attempts to impersonate them. Additionally, the combination of anonymous credential proofs and the CAPTCHA test, protects the VoIP system from the unauthorised use of credentials and from users that misuse their credentials for making SPIT calls. Therefore, this work does not aim at making the CAPTCHA test obsolete through the use of anonymous credential proofs. Using the PrivCAPTCHA approach, callers can assert that they are human users and have certain characteristics, but this is also verified during the CAPTCHA test.

2.2. Discrimination issues concerning CAPTCHA challenges

Traditionally, problems of accessibility to IT applications and services were addressed by adapting their design to the so-called "average or typical user", a feature that actually does not exist. CAPTCHAs had been initially recommended and implemented without taking user needs and (dis)abilities as well as accessibility issues into consideration. However usability and accessibility are seriously affected by most modern visual CAPTCHAs as they pose problems to blind, visually impaired or dyslexic users and, in general, users with disabilities (May, 2005).

Indeed, a CAPTCHA test, which cannot be solved due to the mental or physical disabilities, language, genre, age or even cultural differentiation of the challenged user, interferes with her communication rights (access to and use of IT means) and raises significant discrimination issues (Basso and Bergadano, 2010). A person who cannot respond to a CAPTCHA test on the ground of a disability is discriminated both as subscriber/user of a communication service and as personality, who faces barriers to her communicative interaction with other persons. The use of such a SPIT

detection mechanism impairs her right to free communication and consequently the legally embedded right to receive and impart information (Marias *et al.*, 2007).

The United Nations Convention on the Rights of Persons with Disabilities identifies accessibility as one of its general principles and states that States Parties shall take appropriate measures to promote access for persons with disabilities to new information and communications technologies and systems. In many countries, including the EU countries and the US, legislation in place has to ensure that products and services are accessible and usable by as many users as possible, including people with disabilities and aged persons.

In order to face and/or limit the discriminatory effect of CAPTCHA tests, they should be accessible and usable by all human users, regardless of their cognitive, physical, sensory or cultural characteristics (Fritsch *et al.*, 2010). Apart from having the possibility to switch to a new challenge involving different sensory abilities, the introduction of personalised profiles that take into account the user's diversity, needs and preferences is not only at the core of the inclusive design approach (Fritsch *et al.*, 2010), but seems to be an appropriate response to discrimination concerns.

This approach engages the user in the definition of challenges and tests and considers her needs and abilities. At the very centre of personalised services is the user profile or personal profile, which is a collection of the user preferences and data. However, the personalised CAPTCHA service must be designed in a way that allows the user to use and have access to it, while determining when and who should get knowledge about her preferences and/or disability status (Fuglerud *et al.*, 2009). This requirement derives both from the dignity principle and the privacy rights of individuals.

By definition personalised profiles, i.e., in our case, personalised CAPTCHAs, require collection and use of personal data (age, education level etc) that may also be sensitive (medical data, disabilities, cultural/religion), affecting the privacy rights of the concerned users. By referring to privacy in this paper we focus on the right of the individual to be in control of the information concerning her so as to formulate conceptions of self, values, preferences, goals and to protect her life choices from public control, social disgrace or objectification.

https://www.emeraldinsight.com/doi/full/10.1108/ICS-07-2014-0046

3. PrivCAPTCHA Architecture

In this section the components of the proposed architecture are presented first, comprising the cryptographic building blocks used to achieve the privacy properties and the entities that participate in the system. Then, the resulting functionality is described.

3.1. Cryptographic building blocks

In order to achieve the privacy-preserving attributes of PrivCAPTCHA, we use anonymous credentials and a data management unit as cryptographic building blocks. In this section we provide a high level description of these building blocks, focusing on their attributes and functionality.

3.1.1. Anonymous credentials

Anonymous credentials (Camenisch *et al.*, 2011) allow users to acquire credentials and demonstrate them without revealing their identity. Using the private credential system described in (Camenisch and Pfitzmann, 2007), individuals can use different unlinkable pseudonyms, based on the same credential issued by an identity provider. The private credential system can also provide certified attributes by the identity provider, for the individual to selectively reveal attributes (e.g. their age range, based on their date of birth). Anonymous credentials constitute today an accepted and applicable privacy enhancing technology[1]. In our work we use Idemix[2], an open source anonymous credential system (see implementation in Section 4.1).

The entities participating in an anonymous credential system are *individuals*, *companies* and *trusted third parties* (e.g. government services), which can assume the roles of *issuers*, *recipients*, *provers* and *verifiers.* A credential is created by an issuer for a recipient by executing the *issuing protocol*. The recipient (i.e. the credential owner) can then create a credential proof, to be used by a verifier to verify the validity of the credential (*proving protocol*).

To be able to issue anonymous credentials, an issuer needs to generate a public key pair and create specifications of the structures of the credentials issued. These specifications and the issuer's public key are then published to be used during the proof protocol.

In order to acquire an anonymous credential, a user chooses a master secret key, according to the agreed upon system parameters (bit length, groups to be used). This secret key enables the creation of multiple unlinkable pseudonyms by the user, to be used with different service providers (in our case VoIP services). The issued credential consists of the issuer's public key, the credential structure (necessary for the verification of its validity) and the attribute values.

During the proving protocol, the prover (i.e. the credential owner) creates a proof on behalf of the verifier that proves ownership of a certain credential. The verifier checks the validity of the given proof. Credential attribute values contained in the proof may or may not be revealed to the prover, according to the settings of the proof creation process.

3.1.2. A local data storage and management unit

A data management unit that resides at the owner's side, similar to the Portfolio architecture proposed in (Tasidou and Efraimidis, 2012) is used to manage the user's credentials and certificates. The contents of a user's portfolio include:

- Anonymous credentials, containing verified demographic data and personal characteristics, e.g. age, education level, disabilities.

- Certificates of successful CAPTCHA tests issued by the CAPTCHA service. This transaction history can be used to provide further evidence to the CAPTCHA server that the user is human and non-malicious and can even be used to allow users to pass over the CAPTCHA test for a limited time.

3.2. Entities in PrivCAPTCHA

The entities that participate in the proposed system are the following:

The Identity Provider (IDP). Users obtain their credentials from the IDP, by registering an identifier (e.g. their social security number) and a pseudonym P. The IDP is considered a trusted third party (like a passport authority) that retains the user information together with their pseudonym. The IDP does not need to be a single entity, it can be a distributed service to achieve better service availability and enhanced security.

The User. In our system the users are considered the VoIP service users. All can act both as callers and callees. When acting as callers, their portfolio information can be used to receive personalised CAPTCHA challenges as illustrated in Figure 1.

The CAPTCHA server, which acts as a verifier for the anonymous credential system. Moreover, the CAPTCHA server automatically generates the CAPTCHA challenge, according to the proven user attributes and evaluates the provided answer.

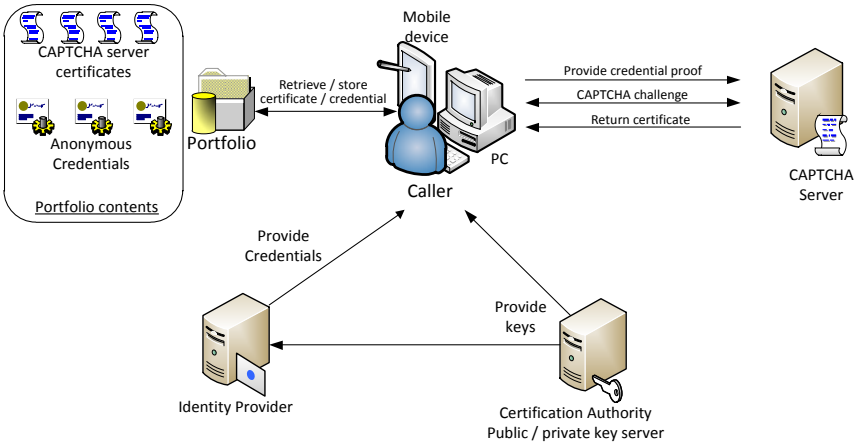

Figure 1: PrivCAPTCHA architecture

The entities of the PrivCAPTCHA system and their interactions are shown in Figure 1. The user's profile is stored at the personal portfolio residing at the user's side. After registering with an IDP and obtaining the anonymous credentials, the caller can prove some attributes to the CAPTCHA server and receive personalised CAPTCHA challenges.

According to the proportionality principle the IDP retains no more data than that strictly required to serve the personalised CAPTCHA service. The IDP entity combines the retained data with a pseudonym in order to protect the identity of the user and it is not allowed to reveal or to use this data for any other purpose, with the exception of law enforcement purposes if required and to the extent that is provided by the respective law. The proposed architecture provides personalised and effective VoIP CAPTCHAs while preserving the privacy and communication rights of the user.

3.3. System functionality

In this section we will describe the general functionalities of the system. Regarding the cryptographic primitives used (see Section 3.1), we adhere to their descriptions as proposed by their authors and we only provide descriptions of their use within the context of our work.

Proving attributes to the CAPTCHA system. After acquiring the anonymous credentials by the IDP the user can begin using them to prove attributes to CAPTCHA services in order to receive personalised challenges. The user needs to prove to the CAPTCHA service that the credential is valid, which verifies that the user has a certain attribute. The verification mechanism is based on efficient zero knowledge proofs (Camenisch and Pfitzmann, 2007). In the context of VoIP

communications, the credential proof can be incorporated into the SIP INVITE message and transmitted seamlessly during the VoIP call (see details in Section 4.3).

CAPTCHA generation and outcome. After receiving and verifying the user's credential proof, the CAPTCHA service generates the appropriate CAPTCHA challenge, according to the user characteristics. The characteristics that the CAPTCHA server takes into account in our example implementation are:

> *Disability status*. Depending on the user's disability, the system can adapt the CAPTCHA challenge to provide a more appropriate challenge for the user's abilities. For example, hearing impaired users may need an elevated volume level for the challenge.
>
> *Language requirements*. Based on the user's native language, the CAPTCHA server can provide the appropriate challenge, either a challenge in the user's language, or some adaptation to facilitate non-native speakers of the challenge language.
>
> *Age*. The age of the user affects the ability to solve tests. If the user is too young or too old, then the CAPTCHA challenge should be adapted accordingly, e.g. to contain less characters to be recognised or to allow more time to solve the test.
>
> Apparently, additional user characteristics can be considered apart from the aforementioned ones, like education level (e.g. literacy), learning disabilities (e.g. dyslexia), etc. The selection of appropriate adaptations for the challenge to facilitate each user characteristic is outside the scope of this work. Our goal is to illustrate the mechanism for the adaptation according to any given user characteristic.

CAPTCHA server certificates. Upon successful completion of a CAPTCHA challenge, the CAPTCHA server sends the user a certificate, attesting that this user did make a legitimate communication. This certificate is stored into the user's portfolio and can be used later by the same user to prove previous legitimate use of the CAPTCHA service.

The above functionalities are prone to misuse and malicious behavior on the part of the user. In Section 3.4 we address the main issues that have been identified for the proposed system.

The communications protocol in PrivCAPTCHA contains the following steps:

1. The user makes a SIP call, containing the credential proof (see Section 4.3).

2. The CAPTCHA server verifies the provided credential proof.

3. The CAPTCHA server generates the appropriate challenge.

4. The user responds to the CAPTCHA challenge.

5. If the challenge response is correct, the CAPTCHA server sends the user a certificate of successful completion.

3.4. Incident response requirements

Designing a system on a user-centric driven security basis requires that the system is robust in the sense that the user is not significantly exposed during a security failure. In the proposed system we have identified the following aspects and requirements for incident response and escalation procedures in the event of a security failure.

Tolerance to false positives. We can in principle consider that false positives carry a minor security impact. The event of a bot successfully answering the audio CAPTCHA challenge will be detected by the destination/callee and the service should maintain the facility for the callee to report/redirect the call for further logging and analysis. Responding to false positives is a good example of active user participation in the security process. Regarding CAPTCHA server certificates, in case of false positives, a revocation procedure can be followed upon receipt of the callee report.

Tolerance to false negatives. Rejecting a legitimate call request after a failed audio CAPTCHA attempt is an event of major significance. Therefore the underlying security parameters are expected to be set on a level where the false negatives are minimised despite the drop in security. Indeed, giving priority to user acceptance over security is part of the user-centric system design practice. In addition, there needs to be a continuous evaluation similar to vulnerability assessment practices. More specifically, as a security administrator must be informed and proactively search for new vulnerabilities affecting the system, the audio CAPTCHA engineer must keep the system up to date with the state of the art research in order to maintain the optimum level of security versus user acceptance.

Tolerance to CAPTCHA server certificate misuse. The certificates provided by the CAPTCHA server can be (un)intentionally misused by users to exploit the system. In case of reported malicious use of these certificates, revocation methods (Camenisch *et al.*, 2011) can be examined.

Correlate system failures with SPIT results. A threat management system should be implemented and the audio CAPTCHA service should be placed in the wider system security context in order to identify threat vectors that may target the CAPTCHA but also exploit the system as a whole.

Reputation management. Reputation mechanisms introduce a number of security issues and should these become part of the audio CAPTCHA service, reputation

misuse should be addressed with well defined escalation procedures. The proposed system can adopt published procedures and controls for reputation management.

4. Implementation and Tests

To demonstrate the feasibility of our proposed system, we created an implementation that integrates the core PrivCAPTCHA functionality into a real-world, open source VoIP application. This implementation allows a credential proof to be transmitted to the CAPTCHA server within a SIP call. For this purpose we created a SIP custom header containing the credential proof for the CAPTCHA server to receive and verify and introduced it into the SIP INVITE message. Additionally, to allow the CAPTCHA server to verify the credential proof, we created a ProofVerifier executable that is called when the SIP message with the custom header is received. In the following sections we describe the developed implementation and the experimental results from its execution.

4.1. The PrivCAPTCHA anonymous credentials

The *Identity Mixer* cryptographic library[3] was used to create the anonymous credentials for the CAPTCHA user and the proof to be sent to the CAPTCHA server. The library implements the anonymous credentials of Camenisch and Lysyanskaya (2001), i.e. the functionality of anonymous authentication for the issuer, client, and service provider.

To create the PrivCAPTCHA anonymous credentials for the implementation we used the Idemix library (Release 2.3.4). We created the appropriate credential structure and proof specification for our application and, utilizing the library functionality, added implementations for the issuance, proof creation and verification methods of the PrivCAPTCHA credentials.

As mentioned in Section 3.3, the credentials created for this implementation describe the following credential-owner characteristics and corresponding enumerated attributes:

- *Disability status*: Hearing Impaired, Blind, Illiterate
- *Native language*: English, other
- *Age group*: Child, Adolescent, Adult, Elderly

Further categories and possible values can be added to describe user characteristics, to suit the needs of each application.

https://www.emeraldinsight.com/doi/full/10.1108/ICS-07-2014-0046

The credential structure used in our implementation is presented in Figure 2, following the credential annotation described in (Bichsel and Camenisch, 2010). The credential information is partitioned into the attributes, defined by a name, issuance mode and type of attribute, and the implementation, where implementation specific information is provided. The enumerated attributes are implemented by assigning a distinct prime to each possible attribute value, according to (Camenisch and Gross, 2008).

```
Attributes{
Attribute { Status, known, type:enum }
   { HearingImpaired, Blind, Illiterate }

Attribute { NativeLanguage, known, type:enum }
   { English, Other }

Attribute { AgeGroup, known, type:enum }
   { Child, Adolescent, Adult, Elderly }

Implementation{
PrimeFactor { Status: HearingImpaired = 3 }
PrimeFactor { Status: Blind = 5 }
PrimeFactor { Status: Illiterate = 7 }
PrimeFactor { NativeLanguage: English = 11 }
PrimeFactor { NativeLanguage: Other = 13 }
PrimeFactor { AgeGroup: Child = 17 }
PrimeFactor { AgeGroup: Adolescent = 19 }
PrimeFactor { AgeGroup: Adult = 23 }
PrimeFactor { AgeGroup: Elderly = 29 }

AttributeOrder { Status, NativeLanguage, AgeGroup}
}
```

Figure 2: PrivCAPTCHA credential structure

In Figure 3 we present an example credential in accordance to the PrivCAPTCHA credential structure (Figure 2):

```
References{
Schema=http://privCAPTCHAdomain.com/credCAPTCHA.xsd
Structure=http://privCAPTCHAdomain.com/CredStructCAPTCHA.xml
IssuerPublicKey=http:// privCAPTCHAdomain.com/exampleIPK.xml
}

Elements{
Signature { A:..., v:..., e:... }
Values { Status:HearingImpaired; NativeLanguage:English;
AgeGroup: Elderly }
}
```

Figure 3: PrivCAPTCHA example credential

To implement the proving protocol, we needed to create a proof specification, shown in Figure 4, for our example credential (Figure 3).

```
Declaration{ id1:revealed:enum; id2:revealed:enum;
             id3:revealed:enum;}
ProvenStatements{
   Credentials{
     randName1: http://privCAPTCHAdomain.com/CredStructCAPTCHA.xml=
        { Status:id1, NativeLanguage:id2, AgeGroup:id3 }
     }
}
```

Figure 4: PrivCAPTCHA proof specification

The proof specification contains:

- Credentials that the user proves ownership of.
- Identifiers for the values included in the proof. Identifiers are assigned to attributes that are fully or partially revealed.
- Attribute types of each identifier, which need to match during the proof protocol.
- Constants that can be assigned to identifiers.

4.2. Jitsi - Open source VoIP application

To integrate the SIP custom header containing the credential proof into the VoIP call, we used Jitsi[4,5], an open source multi-platform audio/video Internet phone application. It supports several instant messaging and telephony protocols, including the Session Initiation Protocol (SIP) used in VoIP networks. Jitsi and its source code are released under the terms of the LGPL. Jitsi is mostly written in Java and, among others, it uses the JAIN-SIP protocol stack for SIP support.

For the purpose of our implementation we downloaded the Jitsi v2.2.4603.9615 source snapshot and added code to introduce the custom header containing the credential proof into the SIP INVITE message.

4.3. SIP custom header

For the credential proof to reach the CAPTCHA server, we introduced the *CredentialProof* custom header into the SIP INVITE message sent by Jitsi during the SIP call. To achieve that, the class CredentialProof was created, extending the ParametersHeader class from the JAIN-SDP library [6], which implements the parameters setting functionality of the SIP headers. Using the *createRequest* method of the *SipMessageFactory* class in the net.java.sip.communicator.impl.protocol.sip package, the custom header was appended into the SIP INVITE message header.

Using this implementation, each time a Jitsi user makes a call, the CredentialProof header is transmitted to the recipient (in our case the CAPTCHA server). The CredentialProof header is shown in Figure 5, where the SIP call performed by Jitsi is captured with the Wireshark packet analyzer[7].

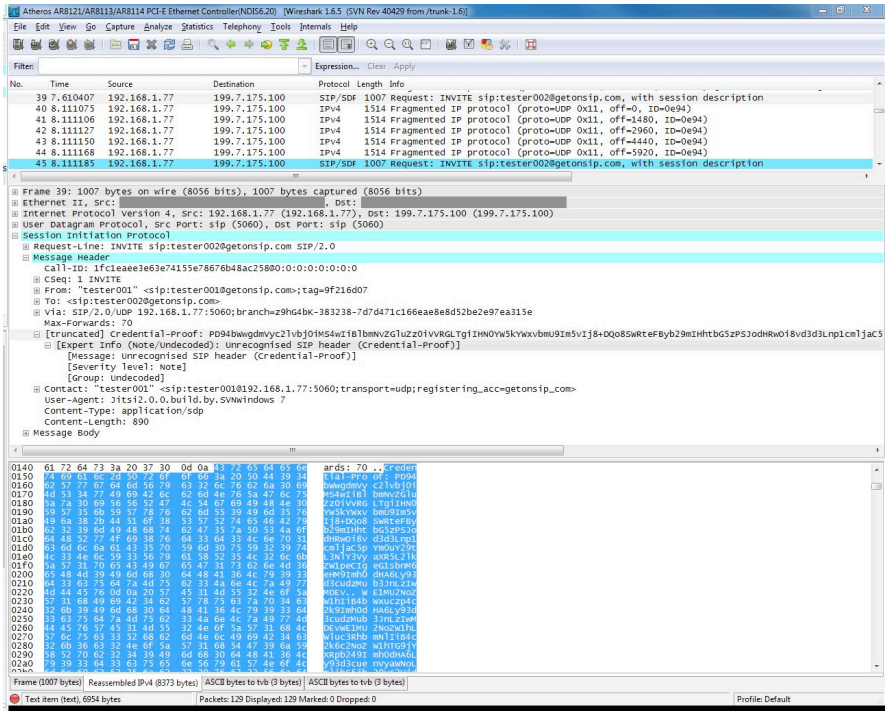

Figure 5: CredentialProof custom header captured with Wireshark during SIP call

4.4. Experimental network environment

The network environment (Figure 6) that we have implemented consists of the following fundamental entities:

An Asterisk server, which was based on AsteriskNow[8], a widespread open source SIP server implementation, also used for VoIP PBX. The provided API of the server supports easy manipulation of SIP headers and allows storing useful metadata in the call-records database. The VoIP PBX runtime environment offers administration access via command line or through the FreePBX web-based application over an Apache server and includes a MySQL database, that stores operational parameters, such as SIP extensions, voice trunks, call records, etc. The implemented server has been customised in order to register users, redirect SIP messages and establish calls. The PC used for setting up the Asterisk server was a Pentium 4, 2.8GHz with 2GB RAM.

A CAPTCHA service. The audio CAPTCHA was implemented as a separate service for efficiency reasons, since the computational resources needed for such a module are considerable. The service was implemented on the AsteriskNow software as well. The basic algorithm was developed using the PHP class Asterisk Gateway Interface (PHPAGI[9]), which interacts with the AsteriskNow software to provide audio CAPTCHAs as a standalone service. The CAPTCHA service:

1. receives the SIP message,
2. extracts the values of the custom header,
3. passes the values to the PHPAGI module,
4. identifies the characteristics of the user asked to solve the CAPTCHA,
5. selects and "plays" the appropriate audio CAPTCHA file based on the proven characteristics,
6. validates the answer and either sends the decision to Asterisk server or it re-sends a new CAPTCHA.

Various VoIP callers. These callers are programmed to make calls to the VoIP service clients. In our scenarios, they are redirected through the CAPTCHA service. The exact number of the external callers depends on each use case/scenario.

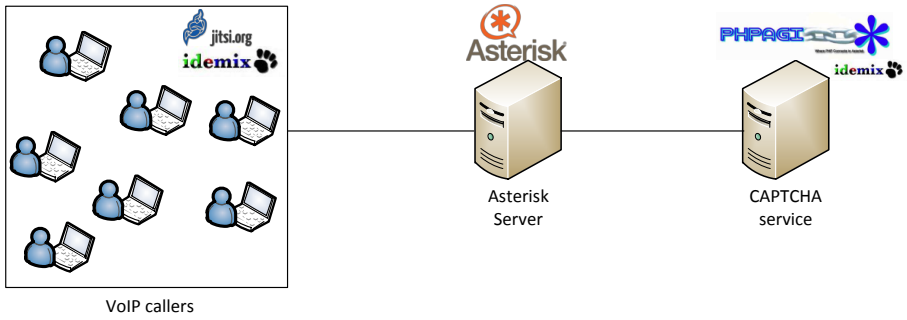

Figure 6: Experimental network environment

4.5. Proposed scenarios and results

For the experimental process we created two scenarios. In both scenarios we measured the time needed for the CAPTCHA server to select the appropriate audio file and set the right playback attributes (volume, number of retries, etc.).

The first scenario was to have a single call initiated and measure the aforementioned needed time. The time needed without the PrivCAPTCHA validation was 3.6ms and with it was 7.8ms. Even though the time has been doubled it still stays extremely low, so there is no significant overload to the system.

https://www.emeraldinsight.com/doi/full/10.1108/ICS-07-2014-0046

The second scenario consisted of 20 external clients, which initiated new calls. The SIP calls were generated randomly, but there was a limit of a maximum of 20 simultaneous calls. The average time interval between the calls was 100ms. The total number of calls was 460. Finally, it should be stated that the calls were terminated 10 seconds after the call establishment, while playing the CAPTCHA audio file. This means that the CAPTCHA service has to keep the calls established either for 10 sec or until the CAPTCHA timeout is reached, which is based on the chosen characteristics. This scenario was created using the SIPp[10] call generator.

In Figure 7 we present the results of the second scenario. The min and max values of each measured variable are represented by the top (⊤) and bottom (⊥) bars. The mean value and one standard deviation from it are represented by the (-) bar and the greyed box respectively. The first column depicts the time need for the proof to be verified by the CAPTCHA server, i.e. the delay caused by the PrivCAPTCHA addition in the system. The second column represents the total time needed for the request to be processed and the third column the time needed for the CAPTCHA challenge to be generated.

The results show that the proof verification process requires approximately 300ms, with relatively narrow deviation, which we consider to be low enough to make this addition a feasible option. Additionally, the efficiency of this verification can be further improved via a server to amortise Java Virtual Machine startup costs, which in this experiment were calculated to be approximately 100ms.

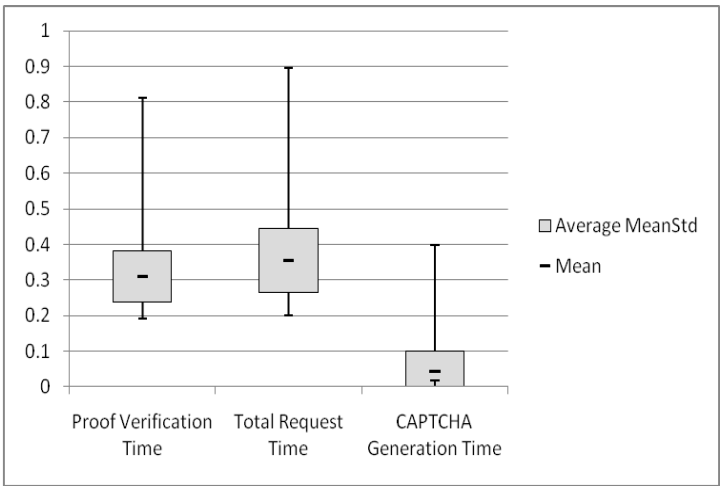

Figure 7: Experimental performance results (in seconds)

This experiment was conducted to demonstrate primarily the feasibility of the proposed system, not its efficiency. Therefore, the times recorded indicate an upper bound on the processing time. In case PrivCAPTCHA were to be used in a

production CAPTCHA service, the efficiency of the system can be expected to be enhanced, by removing overheads and setting up more than one CAPTCHA servers.

Nevertheless, it is a fact that the addition of PrivCAPTCHA introduces an additional computationally intensive task to the SIP communication process, decreasing the CAPTCHA server's maximum capacity for concurrent connections. This can be exploited to enhance the effectiveness of attacks against the SIP protocol, an existing problem of VoIP communications (Dantu *et al.*, 2009, Keromytis, 2012). Since the additional header introduced is created according to the SIP protocol specification, the same countermeasures and policies used to protect conventional VoIP systems can be employed to protect the PrivCAPTCHA system (Soupionis and Gritzalis, 2011, Geneiatakis *et al.*, 2007, Ehlert *et al.*, 2010). These can reduce the amount of malicious messages that reach the CAPTCHA server and invoke the header verification process.

For example, possible countermeasures include SIP message evaluation mechanisms, such as a SIP message parser, which will evaluate the CredentialProof header content. In case of an unsophisticated attack, during which the header is used without maintaining the valid xml schema of the proof, the message will be filtered by the SIP parsing mechanism, therefore never reaching the CAPTCHA server. During a more sophisticated attack, wherein the required xml schema of the proof is valid and only the proof components are not, or a valid CredentialProof is replayed, the proof validation mechanism will be invoked, increasing the server's computational load and possibly leading to service downtime. In this case logging and blacklisting mechanisms can be used to block the malicious calls and server resource monitoring can be used to disable the CAPTCHA service and prevent incontrollable VoIP service downtime when resources are depleted until the attackers are blacklisted.

5. Discussion and Conclusions

In this work, we propose a user-centric, privacy-preserving VoIP CAPTCHA adaptation approach. The PrivCAPTCHA architecture combines existing cryptographic technologies, which provide strong privacy guarantees, utilised under a new context. The proposed system aims at providing an improved CAPTCHA service that is more appropriate for and fair to the human users. Descriptions of the proposed system functionalities are provided and an experimental implementation is carried out within the VoIP protocol. Experimental results show that the utilization of cryptographic tools introduces an additional computational overhead into the audio CAPTCHA application. We consider this overhead to be tolerable for modern computational platforms possibly combined with appropriate performance optimization techniques. However, this overhead can be exploited to increase the efficiency of attacks against the SIP protocol, therefore appropriate countermeasures should be employed to decrease these negative effects. Moreover, although we mainly consider CAPTCHA challenges for VoIP calls in this work, we believe that this idea can be useful for providing a general mechanism for CAPTCHA adaptation according to the user's characteristics.

https://www.emeraldinsight.com/doi/full/10.1108/ICS-07-2014-0046

6. Notes

1. Identity Mixer, http://www.zurich.ibm.com/idemix/details.html

2. Identity Mixer - Usage, http://www.zurich.ibm.com/idemix/usage.html

3. Identity Mixer cryptographic library, https://prime.inf.tu-dresden.de/idemix/

4. Jitsi features, https://jitsi.org/Main/Features

5. Jitsi – Wikipedia article, http://en.wikipedia.org/wiki/Jitsi

6. JSIP: Java API for SIP signalling, https://jsip.java.net/

7. Wireshark, http://www.wireshark.org/

8. AsteriskNow, http://www.asterisk.org/downloads/asterisknow

9. PHPAGI, http://phpagi.sourceforge.net/

10. SIPp, http://sipp.sourceforge.net/

7. References

Ahn, L. von, Blum, M. and Langford, J. (2004), "Telling humans and computers apart automatically", *Communications of the ACM,* Volume 47, Number 2, pp. 56-60.

Basso, A. and Bergadano, F. (2010), "Anti-bot Strategies Based on Human Interactive Proofs", in Stavroulakis, P. and Stamp, M. (Eds.) *Handbook of Information and Communication Security,* Springer, Berlin / Heidelberg, pp. 273-291.

Bichsel, P. and Camenisch, J. (2010), "Mixing Identities with Ease", in Leeuw, E., Fischer-Hübner, S. and Fritsch, L. (Eds.) *Policies and Research in Identity Management,* Springer Berlin Heidelberg, pp. 1-17.

Bigham, J. P. and Cavender, A. C. (2009), "Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use", in *Proceedings of the 27th international conference on Human factors in computing systems*, Boston, MA, USA, 2009, pp. 1829-1838.

Bursztein, E., Bethard, S., Fabry, C., Mitchell, J. C. and Jurafsky, D. (2010), "How good are humans at solving CAPTCHAs? a large scale evaluation", in *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, Oakland, California, USA, 2010, pp. 399-413.

Camenisch, J., Dubovitskaya, M., Kohlweiss, M., Lapon, J. and Neven, G. (2011), "Cryptographic Mechanisms for Privacy", in Camenisch, J., Fischer-Hübner, S. and Rannenberg, K. (Eds.) *Privacy and Identity Management for Life,* Springer, Berlin / Heidelberg, pp 117-134.

Camenisch, J. and Gross, T. (2008). "Efficient attributes for anonymous credentials", in *Proceedings of the 15th ACM conference on Computer and communications security.* Alexandria, Virginia, USA: ACM, pp. 345-356.

Camenisch, J. and Lysyanskaya, A. (2001), "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation", in Pfitzmann, B. (Ed.) *Advances in Cryptology — EUROCRYPT 2001,* Springer Berlin Heidelberg, pp. 93-118.

Camenisch, J. and Pfitzmann, B. (2007), "Federated Identity Management", in Petković, M. and Jonker, W. (Eds.) *Security, Privacy, and Trust in Modern Data Management,* Springer, Berlin / Heidelberg, pp. 213-238.

Dantu, R., Fahmy, S., Schulzrinne, H. and Cangussu, J. (2009), "Issues and challenges in securing VoIP", *Computers & Security,* Vol. 28, No 8, pp. 743-753.

Deswarte, Y. and Gambs, S. (2010), "A Proposal for a Privacy-preserving National Identity Card", *Transactions on Data Privacy,* Vol. 3, No 3, pp. 253-276.

Diaz, C. and Preneel, B. (2007), "Accountable Anonymous Communication", in Petković, M. and Jonker, W. (Eds.) *Security, Privacy, and Trust in Modern Data Management,* Springer, Berlin / Heidelberg, pp. 239-253.

Ehlert, S., Geneiatakis, D. and Magedanz, T. (2010), "Survey of network security systems to counter SIP-based denial-of-service attacks", *Computers & Security,* Vol. 29, No 2, pp. 225-243.

El Sawda, S. and Urien, P. (2006), "SIP Security Attacks and Solutions: A state-of-the-art review", in *Proceedings of the 2nd International Conference on Information & Communication Technologies: From Theory to Applications*, Damascus, Syria, 2006, pp. 3187-3191.

Fritsch, L., Fuglerud, K. and Solheim, I. (2010), "Towards inclusive identity management", *Identity in the Information Society,* Vol. 3, No 3, pp. 515-538.

Fuglerud, K., Reinertsen, A., Fritsch, L. and Dale, O. (2009), "Universal design of IT-based solutions for registration and authentication", Tech. report: DART/02/09, Norwegian Computing Center, Oslo, 2009.

Geneiatakis, D., Kambourakis, G., Lambrinoudakis, C., Dagiuklas, T. and Gritzalis, S. (2007), "A framework for protecting a SIP-based infrastructure against malformed message attacks", *Computer Networks,* Vol. 51, No 10, pp. 2580-2593.

Keromytis, A. D. (2012), "A Comprehensive Survey of Voice over IP Security Research", *Communications Surveys & Tutorials, IEEE,* Vol. 14, No 2, pp. 514-537.

Lazar, J., Feng, J., Brooks, T., Melamed, G., Wentz, B., Holman, J., Olalere, A. and Ekedebe, N. (2012). "The SoundsRight CAPTCHA: an improved approach to audio human interaction proofs for blind users", in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* Austin, Texas, USA: ACM, pp. 2267-2276.

Marias, G. F., Dritsas, S., Theoharidou, M., Mallios, J. and Gritzalis, D. (2007), "SIP Vulnerabilities and Anti-SPIT Mechanisms Assessment", in *Proceedings of 16th International Conference on Computer Communications and Networks*, Honolulu, Hawaii, USA, 2007, pp. 597-604.

May, M. (2005). "Inaccessibility of CAPTCHA. Alternatives to visual Turing tests on the Web". *W3C Working Group Note, November 2005*, available at: http://www.w3.org/TR/turingtest/.

Poller, A., Waldmann, U., Vowe, S. and Turpe, S. (2012), "Electronic Identity Cards for User Authentication; Promise and Practice", *IEEE Security & Privacy,* Vol. 10, No 1, pp. 46-54.

Soupionis, Y. and Gritzalis, D. (2010), "Audio CAPTCHA: Existing solutions assessment and a new implementation for VoIP telephony", *Computers & Security,* Vol. 29, No 5, pp. 603-618.

Soupionis, Y. and Gritzalis, D. (2011), "ASPF: Adaptive anti-SPIT Policy-based Framework", in *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*, 2011, pp. 153-160.

Tasidou, A. and Efraimidis, P. S. (2012), "Using Personal Portfolios to Manage Customer Data", in Garcia-Alfaro, J., Navarro-Arribas, G., Cuppens-Boulahia, N. and De Capitani di Vimercati, S. (Eds.) *Data Privacy Management and Autonomous Spontaneous Security,* Springer, Berlin / Heidelberg, pp. 141-154.

Walsh, T. J. and Kuhn, D. R. (2005), "Challenges in securing voice over IP", *IEEE Security & Privacy,* Vol. 3, No 3, pp. 44-49.

Yan, J. and El Ahmad, A. S. (2008). "Usability of CAPTCHAs or usability issues in CAPTCHA design", in *Proceedings of the 4th symposium on Usable privacy and security.* Pittsburgh, Pennsylvania: ACM, pp. 44-52.