PRIVACY PROTECTION FOR USER AUTHENTICATION

Jan Hajny, Tomas Pelka and Vaclav Zeman
Department of Telecommunications, Faculty of Electrical Engineering and Communication
Brno University of Technology, Purkynova 118, Brno, Czech Republic

Keywords: Privacy, Anonymity, Security, Protocol, Authentication, Identity.

Abstract: The paper deals with Internet user privacy. We focus on the protection of user identity during an Internet service use. We start with an overview of systems for the anonymous channel creation. Such channels are necessary as any solution for the identity protection will have to work with the TCP/IP protocols used in the Internet. In the second part of the paper we argue about the need for an anonymous authentication introduction. Such a service is almost missing in the current Internet. We show how the concept of anonymous authentication could improve privacy and identity protection. Finally we introduce our concept for anonymous authentication with the feature of a malicious user detection.

1 PRIVACY AND IDENTITY PROTECTION

The problem of privacy protection is not limited to the Internet environment only. We use many electronic services every day. It is important to realize that every time we use a service we release some information about us. The examples include employee IDs, library cards, canteen cards and any personal web services like an email, Internet discussions or social sites. It is obvious that more and more services can be used electronically and that the Internet is becoming to be a more powerful environment than any time before. The analysis of current use of authentication can be found e.g. here (Hajny et al., 2009).We leave traces of our Internet behavior and there are many reasons why we should care about how much information we release. The obvious problem is identity theft. In that case someone else misuses released information to act as us on the Internet. We can imagine how serious damage it could cause in the case of Internet banking or the communication with public authorities or employers. The other example of a threat on our digital identity is a behavior scanning. We can see that every time we use a search engine or some e-shop application we get customized results based on a previous usage. It means that our search or buying history is stored somewhere and analyzed for further purposes. But do we really care about more focused advertisements, behavior scanning or movement tracking? Are we too paranoid when thinking about someone analyzing what we do on the Internet? Obviously these questions are appealing for more researchers as many solutions for anonymous surfing emerge.

The paper organization is as follows. We identify and analyze available tools for anonymous routing in the Section 2. The overview includes all practical solutions, namely anonymizer proxies, Crowds, MIXes, Onion Routing and TOR. The concept of anonymous authentication is defined in the Section 3. The Section 4 includes an overview of our approach for anonymous authentication with current results. The Section 5 contains the conclusion with future plans.

2 ANONYMOUS ROUTING

There are many solutions to anonymous routing in TCP/IP available. We included the practical ones, which can be used in practice, to our analysis.

2.1 Proxies

The anonymizer proxies are the simplest solutions. The whole group can be represented by the Anonymizer Internet service (www.anonymizer.com) which works with the mechanism of a proxy server. A client wants to communicate with a server without revealing his identity. That is why he contacts a proxy through a private channel first. All data he sends are encrypted in such a way that only a proxy can decrypt. All requests for the server are sent to the proxy

Hajny J., Pelka T. and Zeman V. (2010).
PRIVACY PROTECTION FOR USER AUTHENTICATION.
In *Proceedings of the International Conference on Security and Cryptography*, pages 90-93
DOI: 10.5220/0002981300900093

server which initiates a new connection to the server and resends all data with proxy's source address. The server receives requests but is unable to distinguish the original initiator as the only address he can see is proxy's address. That is why he sends the reply to the proxy. The proxy creates a private channel with the client and resends data from the server to the client. The whole method is described by Figure 1.

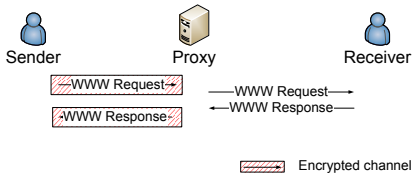

Figure 1: Anonymizer proxy design.

Security of the above described design depends on the proxy entity. It works as a kind of mediator of the communication who resends data between a sender and receiver to keep them separated thus anonymous. On the other hand the proxy must be a trusted service because it is able to link communicating entities together and break privacy. That is the reason why this approach cannot be used in more advanced systems as it is unacceptable to give such a power to a single entity.

2.2 Crowds

A different approach is used by another service called Crowds (Reiter and Rubin, 1998). Security is based on the idea of a "hiding in a crowd" in this case. Traffic traverses through a probabilistic path created by a number of users from the sender to the receiver. The mechanism is based on the alliance of many users of the Crowds service. The sender has a list of participating users available. All users can be used as a relay to resend other's traffic to a receiver. All users send traffic directly to the receiver with a probability of p. With the remaining probability $1-p$ the user sends traffic to a random user of Crowds. That is why data sent by the sender do not go directly to the receiver but to a random Crowds user with probability $1-p$. The random user then sends data to the receiver with p probability and to another random Crowd user with probability $1-p$. That is why a random path through the crowd is generated. No one who intercepts traffic is able to distinguish whether the source address of a captured packet belongs to an original sender or to just a randomly chosen relay. The mechanism can be illustrated by Figure 2. In the picture the requests go as follows: user 1-6-4-receiver2, user 5-2-7-receiver 3 and third request directly from the user 3 to the receiver 1.

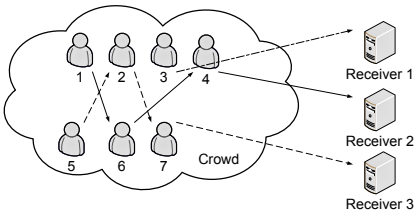

Figure 2: Crowds design.

Crowds use symmetric cryptography and are resistant to the attack of a malicious subgroup of users. Like all solutions in our overview, also Crowds are not resistant to the global adversary who is able to intercept all traffic. Security depends on the average number of nodes between the sender and receiver thus on the parameter p. The more relays are put between the sender and receiver the higher is the probability that a malicious user intercepts a packet with an original sender address inside. Of course there is the price of communication overhead because data must be sent k-times in the case of $k+1$ users in a path. The main advantage to the anonymizer proxies is that users do not have to trust a single entity.

2.3 Web MIXes

Web MIXes (Berthold et al., 2001) is a system based on the mixes mechanism introduced in (Chaum, 1981). The topology is different from the previous examples but combines them to some extent. It is not a peer-to-peer network where all users use all other users like in the Crowds example but it is not a centralized system like anonymizer proxies either. In this case the sender contacts a given structure of entities called MIXes which scrambles and reorders traffic and sends it to the receiver. The idea is that the MIX structure consists of many MIX entities administered by different organizations. These MIXes are not under a centralized control and the whole system works if at least one MIX scrambles honestly. The assumption is that the attacker is able to control $n-1$ MIXes at most. The system works with a program installed on sender's computer which intercepts all (but not limited to) http traffic like a proxy server. These data are encrypted and sent in constant size chunks to the first MIX. The MIX receives, decrypts and reorders data from all senders. After the reorder the MIX resends to another MIX and the process continues as long as data go through all mixes. The last MIX sends data to the receiver. The process can be depicted by Figure 3.

Security of the approach is based on the impossibility of linking the output of MIXes to inputs. This is due to the encryption - data come to MIXes encrypted and are released decrypted and reordered. The sys-

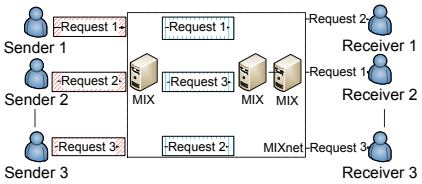

Figure 3: Web MIXes design.

tem needs several layers of encryption as data go through more mixes. To make traffic analysis difficult the system must generate a dummy traffic from all senders for the attacker to be unable to distinguish who is actually sending data and who is quiet. For the future usability it is also very important that Web-MIXes are already implemented and used in practice (https://anonymous-proxy-servers.net/en/).

2.4 Onion Routing and TOR

A similar mechanism is used by Onion Routing (Reed et al., 1998). The sender does not contact the receiver directly but through a proxy software on his computer which sends data through a network of entities called Onion Routers (Fig. 4).

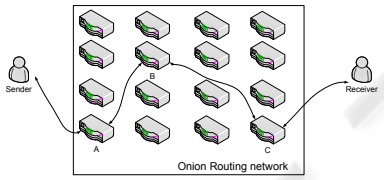

Figure 4: Onion routing design.

Asymmetric cryptography is used to protect data from eavesdropping by malicious Onion Routers. All packets sent by the proxy are encrypted by public keys VK of all routers on the path. The encryption must include next hop headers (*hdr*) - the encryption for a three router situation is described by Equation 1.

$$packet = E_{VK_A}(hdr + E_{VK_B}(hdr + E_{VK_C}(req))) \quad (1)$$

Every router then peels off a layer of encryption as the packet goes through it. It learns a next hop IP and an encrypted payload to send. That is the reason why routers know only its predecessor and successor because all IP addresses of remote routers on the path are encrypted. Security of Onion Routing is based on used asymmetric cryptography and randomness of the path. The routers must be chosen randomly and at least one of them must be honest.

TOR. The practical implementation of Onion Routing is called TOR (Dingledine, 2009). It uses the same principle to choose a random path through routers and to encrypt data with an encryption layer for every router. The difference is in cryptography as TOR uses symmetric cryptography for data encryption and asymmetric cryptography for key establishing. Before actual data are sent through a random path the proxy establishes a symmetric key using a Diffie-Hellman protocol with every router on the path. Then the proxy encrypts data with the established key in a similar fashion as described in the Onion Routing section and sends to the first router. The peeling procedure is the same as for original Onion Routing. The difference is in efficiency as symmetric encryption is much faster in a practical implementation. Security depends on the same factors as in Onion routing, namely cryptography security and at least some honest routers. The TOR design can be used for any protocol and thanks to efficiency and low latencies it is a one of most popular solutions. In our more advanced structures we chose TOR as a solution for anonymous channels.

3 ANONYMOUS AUTHENTICATION

The mentioned solutions provide mechanisms for anonymous routing. We would like to add the feature of anonymous authentication. We introduce a concept which keeps all properties required by service providers but adds anonymity features for service users. Our goal is to keep clients responsible for their acts but reveal their identity only in the case of rule violations. Using our concept the users stay anonymous as far as they obey system rules. On the other hand if they break system rules they can be uniquely identified. The concept with such properties is called Anonymous Authentication and has following requirements:

- Completeness: a valid user must be always accepted.
- Soundness: an other than valid user must be always rejected.
- Anonymity: the verification protocol must release no information about user identity.
- Spread traceability: the user identity must be released but only in the case of rule breaking.

The concept of anonymous authentication can be easily imagined as a group authentication where the group includes all valid users of the system. The verifier is able to check the membership in a group (therefore user's right to use the system) but concrete

identity stays hidden. The traceability is required for a user responsibility. It is in a direct contrast to anonymity and that is why we want it to be spread among more entities which must cooperate to reveal a malicious user identity. Our assumption is that more entities must agree to break anonymity thus the traceability is available only in justified cases.

4 OUR CONTRIBUTION

We decided to use e-cash systems to provide anonymous authentication. We find the ability to create an anonymous e-coin spendable at the e-shop very useful for authentication. There is also a very important feature of double-spender detection available at e-cash systems. It provides the bank with the possibility to reveal identity of a user who spent an e-coin twice. That is why the bank is always sure that the user either used the coin only once or the bank is able to learn his identity in the case of multiple spending.

Our approach uses an e-coin as an authentication token. The user must provide the verifier with a valid e-coin (token) and the knowledge of its construction. If everything is accepted by the verifier the user is successfully authenticated because the token is valid thus released by the administrator of tokens who release tokens only to valid users (similarly the bank releases coins only to users with bank accounts). The completeness and soundness features are directly fulfilled by e-cash system properties. No user identity is released as the e-coin (token) is unlinkable to the user. The only problem is with the spread traceability. We solved the problem by using the double spender detection feature. The user uses the e-coin (token) at a newly defined public authority entity which signs the token. Then the user uses the token for authentication for the second time. That is why he is a double spender so his identity can be revealed but only if the public authority and the verifier cooperate. Neither the verifier nor public authority is able to break user anonymity alone.

Currently we are working on a scheme described in this paper. We use Σ-protocols (Cramer, 1996) as cryptographic primitives and that is why we get very good efficiency. We expect the authentication protocol to be able to run in a smartcard environment.

5 CONCLUSIONS

The purpose of this paper is to point out to insufficient privacy in the Internet and to introduce tools for improving user privacy. We distinguished two steps for providing user anonymity. The first one is anonymous routing which give users the possibility to anonymously communicate. There are practical tools to use. We expect the TOR protocol to be the choice for most implementations as it is both secure and usable for multipurpose traffic. The second step for providing anonymity is unsolved yet. We propose the concept of anonymous authentication which allows users to use not only open services but also services which require some form of user authorization. According to our future plans we would like to implement more features to our anonymous authentication scheme. Our goal is to provide provable security on a smartcard platform.

ACKNOWLEDGEMENTS

Sponsored under the National Program of Research II by the Ministry of Education, Youth and Sports of the Czech Republic in 2C08002 Project - KAAPS Research of Universal and Complex Authentication and Authorization for Permanent and Mobile Computer Networks.

Jan Hajny is a holder of the Brno Stipend for Talented Doctoral Students.

REFERENCES

Berthold, O., Federrath, H., and Köpsell, S. (2001). Web mixes. In *International workshop on Designing privacy enhancing technologies*, pages 115–129, New York, NY, USA. Springer-Verlag New York, Inc.

Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90.

Cramer, R. (1996). *Modular Design of Secure, yet Practical Cryptographic Protocols*. PhD thesis, University of Amsterdam.

Dingledine, R. (2009). Tor: anonymity online. World Wide Web electronic publication.

Hajny, J., Pelka, T., and Lambertova, P. (2009). Flexible Authentication Framework. In *2009 INTERNATIONAL CONFERENCE ON NETWORK AND SERVICE SECURITY*, pages 29–33. IEEE.

Reed, M. G., Syverson, P. F., and Goldschlag, D. M. (1998). Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16:482–494.

Reiter, M. K. and Rubin, A. D. (1998). Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1):66–92.