An Adaptive Three-Party Accounting Protocol

Gregory Detal Damien Leroy Olivier Bonaventure
Université catholique de Louvain, 1348 Louvain-la-Neuve, Belgium
{gregory.detal,damien.leroy,olivier.bonaventure}@uclouvain.be

ABSTRACT

Three-party tunnel-based roaming infrastructures may become a future trend to permit mobile users to connect to the Internet when they are not at home. Those solutions takle security issues for both visited networks and mobile users, but require an efficient and scalable accounting protocol. In this paper, we present a lightweight accounting protocol in which the quantity of data that the mobile is allowed to send is gradually increased when cryptographically signed receipts are received.

Categories and Subject Descriptors

C 2.2 [**Computer-Communication Networks**]: Network Protocols; C 2.0 [**Computer-Communication Networks**]: General—*Security and Protection*

General Terms

Security, Design

Keywords

WiFi roaming, Protocol, Accounting

1. INTRODUCTION

With the widespread deployment of Internet capable mobile devices, more and more users are asking for solutions giving them Internet connectivity anywhere at anytime. An easy way to achieve this goal could be to use WiFi networks already deployed in most public and private places. However, such a type of solution raises many security issues for both the visited network and the mobile user. One proposal, fulfilling these security needs, rely one the combination of a three-party authentication and a tunnel creation from the visited to the user's home network [4]. Fig. 1 illustrates the three communicating entities in this specific roaming infrastructure: the mobile user $\mathcal{M}$, the visited network $\mathcal{F}$ and the user's home network $\mathcal{H}$. $\mathcal{M}$ is located in $\mathcal{F}$ network and his packets are directly forwarded through an authenticated tunnel by $\mathcal{F}$ to $\mathcal{H}$. $\mathcal{H}$ as a proxy to the Internet.

In this infrastructure, no convenient accounting solution currently exists to permit $\mathcal{F}$ to charge $\mathcal{H}$ for $\mathcal{M}$'s network resource consumption. $\mathcal{H}$ must rely on accounting information provided by $\mathcal{F}$. However, difficulties may arise if one disagrees on the volume of data transmitted and denies the invoice. To avoid such conflict, cryptographic solutions propose non-repudiation techniques, using a third-party entity and/or ciphering methods [3]. However, these solutions do not scale well to the Internet environment, due to constraints such as limited bandwidth, delay, limited hardware capabilities, etc.

We propose a lightweight accounting protocol to charge $\mathcal{H}$ for the forwarding service provided by $\mathcal{F}$ based on the number of bytes sent by $\mathcal{M}$. The protocol limits the the risk for $\mathcal{F}$ not to be paid despite the possible packet loss. In this paper, we only focus on the tunnel-based roaming infrastructure. However, the protocol could also be used in other contexts involving a third party relay, e.g., to charge two communicating clients using a SIP proxy. Due to space limitations we only discuss the unidirectional data sent from $\mathcal{M}$ to $\mathcal{H}$. Nevertheless, it can also be easily employed in the opposite direction.

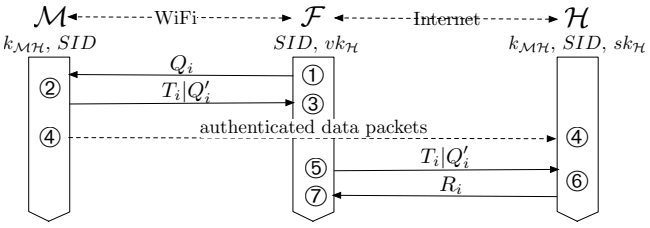

Figure 1: The accounting protocol in an unidirectional environment (all messages are assumed to be authenticated)

We suppose that $\mathcal{M}$ and $\mathcal{H}$ trust each other and want to transmit data in both directions while paying as less as possible and without being suspended by $\mathcal{F}$ for not agreeing with the accounting information. $\mathcal{F}$ forwards the data and wants to be paid as much as possible without being suspended by $\mathcal{M}$ or $\mathcal{H}$ for over-billing. However, it is intrinsically impossible to distinguish an actual loss across the Internet from dropping packets. $\mathcal{F}$ can always pretend that it has transmitted all the data even though it dropped them. Likewise, $\mathcal{H}$ could claim that it did not receive anything from $\mathcal{F}$ even if it is fallacious. We consider $\mathcal{F}$ as selfish, i.e., it may drop packets or send faulty bills to increase its revenue, but not malicious, it does not intercept nor redirect packets. If one party disconnects or claims that the other one has violated the agreement, the data volume not charged must remain below a certain threshold.

Fig. 1 shows, above each box, the crypto materials known

CoNEXT'09 Student Workshop, December 1, 2009, Rome, Italy.
ACM 978-1-60558-636-6/09/12.

after the session establishment that precedes the accounting protocol execution [4]. First, a symmetric key $k_{\mathcal{MH}}$ is used to authenticate the accounting information between $\mathcal{M}$ and $\mathcal{H}$. Second, a unique session id SID is used to identify the roaming session. Third, an asymmetric key pair $(sk_{\mathcal{H}}, vk_{\mathcal{H}})$ is used by $\mathcal{H}$ to generate a signature with $sk_{\mathcal{H}}$ which can be verified at any time with $vk_{\mathcal{H}}$. We also assume that $\mathcal{M}$ and $\mathcal{H}$ have established an authenticated tunnel (e.g., using IPSec AH) so that $\mathcal{F}$ is not able to modify the packets sent by $\mathcal{M}$ nor inject forged packets.

2. ACCOUNTING PROTOCOL

The main idea of our protocol (illustrated on Fig. 1), is to use a slow-start approach, similar to the one used in TCP, to increase the allocated bandwidth according to the trust-level of $\mathcal{F}$ in $\mathcal{H}$ and $\mathcal{M}$. This iterative mechanism ensures that no party can be scammed for more than a few bytes. In practice, at startup, $\mathcal{F}$ only allows to forward a small amount q_0 of data. When it obtains a receipt, i.e., a non-repudiable proof that it will be paid, for this amount, it allows $\mathcal{M}$ to send q_1 bytes, with $q_1 > q_0$ and so on. To avoid replay attacks and to permit using the latest receipt as the ultimate proof, the protocol uses Q_i instead of q_i, defined as $Q_i = \sum_{j=0}^{i} q_j$. The following, complementary to Fig. 1, describes the messages exchanged during one iteration of the protocol.

① Upon reception of a receipt for the $i-1^{\text{th}}$ iteration, $\mathcal{F}$ chooses q_i, the new amount of bytes it agrees to forward for this iteration. It sends Q_i.

② $\mathcal{M}$ chooses q'_i, the number of bytes it agrees to buy for the current iteration, $Q'_i \leq Q_i$. It sends to $\mathcal{F}$ the value Q'_i and a ticket proving this commitment. This ticket, T_i is computed using a Message Authentication Code (MAC) function on Q'_i and the SID as value and $k_{\mathcal{MH}}$ as key. It is used as a proof for $\mathcal{H}$ that $\mathcal{M}$ agrees on the quantity.

③ Upon reception of $T_i|Q'_i$, $\mathcal{F}$ stores them and starts forwarding packets until Q'_i has been consumed. It measures $Q_i^{\mathcal{F}}$, the amount of data it forwards to $\mathcal{H}$.

④ $\mathcal{M}$ starts sending data. At the same time, $\mathcal{H}$ counts the amount of data it receives from $\mathcal{M}$, this value is denoted $Q_i^{\mathcal{H}}$.

⑤ Once $Q_i^{\mathcal{F}} \geq Q'_i$ bytes have been transmitted, $\mathcal{F}$ requests a receipt from $\mathcal{H}$ by sending the ticket.

⑥ $\mathcal{H}$ must first check whether the reception ratio is acceptable. It verifies that $\frac{Q'_i - Q_i^{\mathcal{H}}}{Q'_i}$ is lower than a pre-defined value. If the ratio is acceptable, $\mathcal{H}$ sends a non-repudiable receipt R_i back to $\mathcal{F}$, by applying an signature on Q'_i and the SID. Otherwise, $\mathcal{H}$ disconnects without validating this iteration of the protocol.

⑦ $\mathcal{F}$ always stores the latest (R_i, Q'_i, SID) it received. It is the non-repudiable proof that $\mathcal{M}$ sent at least Q'_i bytes. In the next iteration, q_{i+1} can be greater than q_i since the confidence level of $\mathcal{F}$ in $\mathcal{M}$ and $\mathcal{H}$ has increased.

In practice, to avoid traffic blocking between steps ⑤ and ⑦, $\mathcal{F}$ sends, in advance, the Q_{i+1} message at step ⑤. It means for $\mathcal{F}$ that it can be scammed of the $q_i + q_{i+1}$ iteration values in the worst case, i.e., if $\mathcal{H}$ does not send T_i before the end of the $i+1^{\text{th}}$ iteration. This is why $\mathcal{F}$ must tune the values of Q_i to its needs and its trust level in $\mathcal{M}$ and $\mathcal{H}$.

3. RELATED WORK

A lot of work has been done on the subject of fair non-repudiation of exchange, i.e, in which no party has any advantage over the other ones in any case [3]. These protocols are mainly based on a trusted third-party and are purely cryptographic, they do not scale well to a real Internet environment.

Hasan *et al.* presented a simpler roaming approach of non-repudiation [2] that can be applied in our architecture. Their algorithm is executed after each session and increases the delay at each session ending as well as the loss of money in case of dispute.

Goldberg *et al.* proposed a monitoring technique in presence of a man-in-the-middle that tries to bias measurements [1]. They described how to represent a set of packets and compare them on both sides. Although their solution could be used in the accounting domain, it is less satisfying than ours in terms of security and efficiency. This is mainly due to the three-party infrastructure that induces strong hypothesis and permits us to relax the problem. For instances, we focus on a point-to-point environment and the accounting is enabled from the first to the last packet.

4. CONCLUSION

Three-party infrastructures such as the one presented in this paper may become a future trend to permit mobile users to connect to the Internet when they are not at home. However, such infrastructures require efficient accounting protocols that can scale. For this purpose, we present a protocol in which the quantity of data that the mobile is allowed to send is gradually increased when receiving cryptographically signed receipts. We are currently working on the integration of our protocol into an existing infrastructure. In further works, we extend this protocol to measure other metrics such as duration or QoS. We also evaluate the traffic overhead of the accounting protocol control messages and the effective end-to-end bandwidth obtained.

Acknowledgement

This work is supported by the Belgian Walloon Region under its RW-WIST Programme, ALAWN Project.

5. REFERENCES

[1] S. Goldberg, D. Xiao, E. Tromer, B. Barak, and J. Rexford. Path-quality monitoring in the presence of adversaries. *SIGMETRICS Perform. Eval. Rev.*, 36(1):193–204, 2008.

[2] H. Hasan and B. Stiller. Non-repudiation of consumption of mobile internet services with privacy support. In *Wireless And Mobile Computing, Networking And Communications (WiMob)*, volume 2, pages 1–8, Aug. 2005.

[3] S. Kremer, O. Markowitch, and J. Zhou. An intensive survey of fair non-repudiation protocols. *Computer Comm.*, 25(17):1606–1621, 2002.

[4] M. Manulis, D. Leroy, F. Koeune, O. Bonaventure, and J.-J. Quisquater. Authenticated wireless roaming via tunnels: Making mobile guests feel at home. In *Proc. ACM Symp. on Information, Computer and Communications Security (ASIACCS)*, pages 92–103, Mar. 2009.