Peer-to-Peer Secure and Private Community Based Multimedia Communications

Pantelis A. Frangoudis and George C. Polyzos

Mobile Multimedia Laboratory
Department of Computer Science
Athens University of Economics and Business
Athens 113 62, Greece
{pfrag, polyzos}@aueb.gr

Abstract

We have designed and implemented P2PWNC, a fully distributed, open to all, autonomous WLAN roaming scheme that can be used in a community to provide various multimedia communication services. Here we discuss various security related issues and the support provided by P2PWNC for several aspects of communication and roaming privacy. We also report on aspects of its performance, focusing on the performance penalty of security-related operations. The core P2PWNC scheme assumes that community members are selfish and do not trust each other and uses a secure incentive technique to encourage their contribution. It protects the real-world identities of community providers and clients by relying only on disposable opaque identifiers and does not rely on any authority to resolve disputes or control membership. It could easily complement cellular networks for low-mobility users in metropolitan areas where some Wireless Community Networks provide wide coverage.

1. Introduction and related work

The *WiFi* technology for *Wireless LANs* (WLANs) has become increasingly popular worldwide for implementing hotspots that provide wireless Internet access in campuses and many other public venues. WiFi enabled network interface cards are now becoming standard equipment for mobile devices such as laptops, PDAs and advanced cell phones. Moreover, low-cost wireless access points (APs) are increasingly used even in households, providing wireless coverage for home networks. The popularity, ease of deployment and low cost of WiFi technology, combined with the wide spread of wireline broadband Internet connections (e.g. Cable and DSL lines) can assist in the realization of true ubiquitous Internet multimedia services, such as Voice and Video over IP.

Especially in urban settings, *Wireless Community Networks* [1][12][18] and privately operated WLANs, in some cases, offer nearly complete wireless coverage. Despite the pervasiveness of WLAN signals, though, ubiquitous Internet access is far from a reality. Security concerns of residential WLAN owners, privacy issues of WLAN users, but, most important of all, lack of the proper incentives for not-for-profit wireless Internet sharing account for that.

We have designed the *Peer-to-Peer Wireless Network Confederation (P2PWNC)* [4], a practical incentive scheme that could be used to stimulate participation in wireless communities and fuel ubiquitous wireless Internet access through the private contributions of individual WLAN owners. Our prototype system for WLAN sharing is based on indirect service reciprocity – only WLAN owners who share their bandwidth with others may consume bandwidth when they themselves are mobile.

Work with similar motivation to ours is presented in [2]. WISPs have multilateral roaming contracts and must register with a central authority that maintains reputation records, which are updated with QoS reports submitted by the roamers. There are also a few commercial solutions that deal with WLAN sharing [19][5][10], which usually involve central management and aim at WLAN sharing in a for-profit basis. Also, there many efforts for city-wide and municipality initiated, funded or in general supported wireless networks, with prominent example that of *Wireless Philadelphia* [24], a not-for-profit initiative of the City of Philadelphia to provide wireless Internet access throughout the city.

We have extended our scheme with an architecture that can be directly deployed on top of P2PWNC to offer Internet multimedia services, with secure Voice over IP calls as its main application. Privacy is enhanced using uncertified and disposable user identifiers, while multimedia services are secured by means of VPN technologies. Our position is that, provided wireless coverage is adequate, these services can become a cheap and secure alternative to 2G/3G in citywide areas, wherever the P2PWNC infrastructure is available.

The remainder of this paper is organized as follows. In Section 2 we provide an overview of the P2PWNC principles and protocol and discuss their security properties. In Section 3, we present an architecture for decentralized and secure multimedia communications, as a direct extension of the core P2PWNC scheme, while in Section 4 we study the effects of the proposed architecture to the performance of embedded devices upon which it is designed to operate. A discussion on various security and privacy related issues appears in Section 5, before we conclude in Section 6.

2. The P2PWNC architecture

2.1. Overview and trust model

In prior work [4], we presented the design, principles and underlying algorithms of the core P2PWNC architecture, as well as the P2PWNC protocol. In our scheme, Internet bandwidth is treated as a peer-to-peer resource. Small teams of users, that operate a number of public WLAN APs connected to fixed broadband links (typically Cable/DSL), represent the system's peers. A team *consumes* each time one of its members accesses the Internet through the AP of a different team, and *contributes* when a member of another team uses one of the team's APs.

Peers (teams) are identified by simple public/private key pairs, which are uncertified and free. There is no trusted central certification authority. Teams recruit members by issuing *member certificates*, which allow team members to consume in the name of the team. Such certificates contain the public keys of the team and the member and are signed by the team's private key. We suppose that there is no trust or cooperation among different teams. Within a team, though, it is assumed that there is no anonymity and there is full trust.

Each time a service provision takes place, a digital receipt is issued by the consumer. Digital receipts encode service dept and represent the system's history of transactions. They form a logical receipt graph, which is the input to a *reciprocity algorithm.* This algorithm identifies contributing teams and exposes *free riders*[1] using maximum flow [7][22] techniques on the receipt graph. Simulations presented in [4] have shown that our algorithm can sustain reciprocal cooperation among selfish peers in citywide areas.

Receipts are stored in a decentralized manner, each team having its own receipt repository with finite size. Old receipts get outdated and are being replaced by fresh ones, based on their timestamp[2]. Receipts have the following format:

{Consumer Certificate, Provider Public Key, Timestamp, Weight}$_{\text{Consumer Signature}}$

2.2. The P2PWNC protocol

P2PWNC entities communicate using a simple ASCII-based protocol. During a P2PWNC session, the AP periodically requests that the client signs a fresh receipt, acknowledging the service he has consumed thus far. The session terminates implicitly as soon as the client fails to deliver a receipt in response to a receipt request (which normally happens when a client walks off the AP). Apart from the receipt generation protocol, there is also a "gossiping" protocol [4], which assists in receipt dissemination among the team-local receipt repositories. Each time a client approaches a foreign AP, he can present the AP with a number of receipts that are then forwarded to the receipt repository. Our protocol runs on top of common WLAN equipment, desktop PCs and WLAN-enabled smart-phones [17]. The RSA and ECDSA digital signature algorithms are both supported. Our reference implementation is open-source and available for download from the project's website [15]. The complete specification along with an initial evaluation of the P2PWNC protocol can be found in [6].

The P2PWNC protocol is designed to protect users from numerous security attacks. First, the authenticity and integrity of the receipts exchanged are ensured, since they are digitally signed by the issuer. Second, session hijacking is avoided, as sessions are refreshed by the digital receipts the service consumer periodically signs. Thus, the hijacker cannot sustain a session without access to the private key of the real service consumer (mobile user). Finally, the AP can prevent replay attacks by means of the receipt timestamp mechanism.

[1] Free riders: Users who consume resources without contributing to the community.

[2] The timestamp field denotes the time a P2PWNC session began. The weight field represents the amount of traffic the AP has forwarded on behalf of a visitor during a session.

3. An architecture for secure decentralized voice and video communication

We now present an architecture for secure and decentralized multimedia communication in metropolitan areas, as a direct extension of the P2PWNC scheme. We assume that P2PWNC participants operate trusted VPN gateways at their home networks. To spare users the need for extra equipment, this functionality can be built into the firmware of the wireless APs that users operate in order to participate in the P2PWNC scheme.

Now that users have their trusted VPN gateways set up, whenever they visit APs belonging to other peers, after setting up a P2PWNC session with the visited AP, they can tunnel all their Internet traffic to their home so that the untrusted visited AP cannot intercept it.

Suppose, now, that a roaming user (W1) wishes to place a VoIP call to another P2PWNC user (W2) who is also roaming around another P2PWNC AP. We assume that the two parties are aware only of each other's mobile phone number and that the public IP of their home VPN gateways is dynamic[3]. A mechanism, thus, needs to be in place so that the two parties can discover each other and communicate. For this purpose, in our design, we use the minimal information available, namely the users' GSM mobile phone numbers.

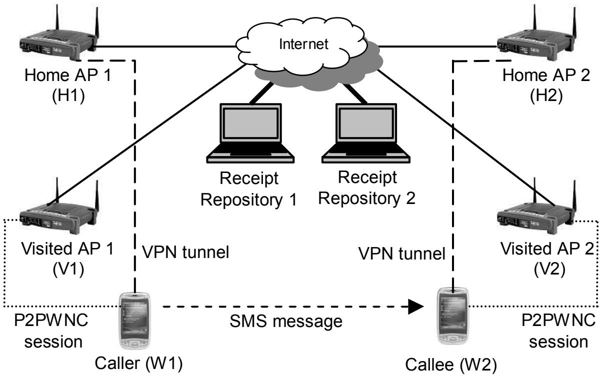

Figure 1. Distributed multimedia call architecture

Figure 1 depicts the process of placing a P2PWNC-based multimedia call. First, W1 and W2 setup P2PWNC sessions with the APs they visit (V1 and V2). Each AP has consulted its own Receipt Repository before granting access to W1 and W2. Then, W1 and W2 setup VPN tunnels to their home networks (H1 and H2 respectively), and, afterwards, W1 initiates a VoIP call to W2 by sending a GSM SMS including the IP address of his home gateway (H1). W2 responds with the multimedia stream, which is tunneled to H2, sent to H1 and, finally, forwarded to W1 through the H1-W1 tunnel. Therefore, without relying on DNS, SIP/H.323 registrars or similar directory services, free and secure multimedia calls can be realized in metropolitan areas, where there is enough WLAN coverage.

4. Performance evaluation

In this section we present a set of experiments that we have conducted to determine how the P2PWNC protocol and the architecture for secure multimedia communications that we have designed on top of it affect the performance of typical networked embedded devices, such as the Linksys WRT54GS [9] wireless router that we have used in our reference testbed. In particular, we test the cost of generating (signing) and verifying digital receipts as well as the routing behavior of a P2PWNC-enabled Linksys AP under heavy routing load when acting also as a VPN gateway. As evident from other work in literature [11][25], the performance penalty both due to VPN-related cryptographic operations and to the space overhead imposed by the security-related information added to the transmitted packets is expected to be significant.

4.1. Testbed and experimental methodology

Our experimental testbed was composed of 14 desktop PCs, two 8-port 100BaseTX Ethernet switches and a Linksys WRT54GS wireless router with the OpenWRT [14] firmware, in which we have included the P2PWNC software. Each switch was used to connect 7 PCs. The exact hardware and software specifications of the equipment used are presented in Table 1.

First, we report the pure CPU time that the generation and verification of a P2PWNC receipt takes on the Linksys box. Second, we measure the TCP throughput achieved by a host (who has set up a P2PWNC session with the AP) over an L2TP/IPsec tunnel that it maintains with the Linksys box in the presence of parallel sessions. We use the ttcp [23] utility and carry traffic over the router's *wired* LAN interface. There are 7 transmitter-receiver pairs. Transmitting hosts are connected (via a switch) to the LAN interface of the wireless router, while receiving hosts are connected to its WAN interface. The Linksys box performs Network Address Translation for the LAN hosts. We used NTP for host synchronization, and then scheduled simultaneous data transmissions using the Linux *crond* scheduler daemon on each host.

[3] This is a typical scenario in the wireless communities that we target. Users typically have Cable/DSL lines connected to their APs and their IP address is dynamically assigned by their ISP.

Table 1. Platform specifications

Characteristic	PC Workstations	Linksys WRT54GS
CPU speed	3.00 GHz	200 MHz
CPU type	Intel Pentium 4	Broadcom MIPS32
RAM	512 MB	32 MB
Storage	2x70 GB HD	8 MB Flash 32 KB NVRAM
Network interfaces	SiS 100BaseTX Ethernet cards	Broadcom integrated 4-port 100BaseTX Ethernet switch 100BaseTX Ethernet WAN interface 802.11g wireless interface
Operating system	Linux kernel 2.6.10 (Knoppix 4)	Linux kernel 2.4.20 (OpenWRT)
Cryptographic Library	OpenSSL 0.9.8b	OpenSSL 0.9.8b

4.2. Setup parameters

We fixed the RREQ interval (the time between two successive receipt requests by the AP to a client) to 5 seconds. This is a reasonable choice considering our expectation that the primary application over P2PWNC will be VoIP; under such an assumption, we expect clients to place VoIP calls of a few seconds to few minutes, so requesting a fresh receipt every 5 seconds ensures that most of the forwarded traffic will be acknowledged by the service consumer. 1024-bit RSA and 160-bit ECC keys have been used. As to the ECDSA algorithm, we have used verifiably random curves over the Fp finite field (in particular, the *secp160r1* and *secp224r1* curves for 160- and 224-bit key lengths respectively [20][21]). We have used the Openswan [13] IPsec implementation and operated it in *tunnel mode*, with *Preshared Keys (PSK)* for authentication. IPsec is used to secure the L2TP tunnels [16] hosts set up with the AP.

4.3. Results

4.3.1. Cryptographic operations. Receipt generation involves the production of the SHA-1 hash of the service provider's public key, the consumer's certificate and the receipt timestamp and weight, and the digital signing of this hash using the consumer's private key. Table 2 shows receipt generation and verification pure CPU times for the two supported digital signature schemes (RSA and ECDSA) for the same security level, carried out on the Linksys box. Equivalent security to 1024 and 2048 bit RSA keys can be achieved using 160 and 224 ECC bit keys respectively [8]. The figures illustrate the fact that private key operations (signatures) are performed faster using the ECDSA algorithm. A commentary on the implications of this fact on the choice of cryptographic parameters in P2PWNC is presented in Section 5.1.

Table 2. Receipt operations (in msec)

Receipt Operation	Security level	
	RSA 1024 / ECC 160	RSA 2048 / ECC 224
Generation	300.6 / 20.3	1529.0 / 23.4
Verification	12.3 / 114.7	37.9 / 135.7

4.3.2. VPN tunneling overhead. We now proceed to determine the cost of using tunnels to secure client communication on the achieved throughput. For comparison, we have included the throughput achieved (1) over pure Ethernet (reference curve), (2) when the Linksys box performs NAT routing (NAT curve), and (3) when LAN hosts have set up P2PWNC sessions with the AP using either RSA or ECDSA receipts.

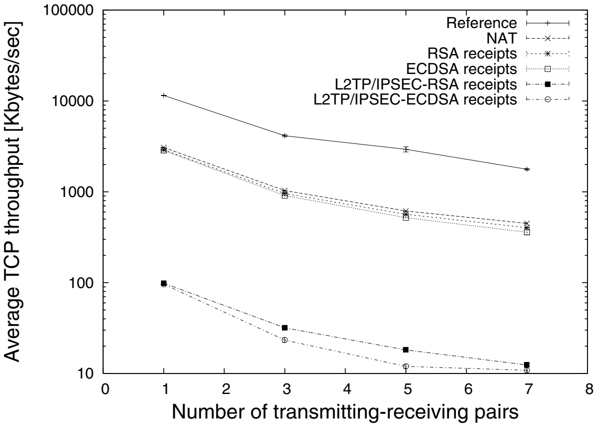

Figure 2. P2PWNC-enabled Linksys WRT54GS router performance as a VPN gateway

The experimental results reveal the performance overhead that the architecture that we proposed in Section 3 will incur. Performance degradation is mainly due to the high protocol overhead of L2TP/IPsec tunneling. Also, CPU cycles are spent for IPsec-related cryptographic operations. As it seems, when the number of concurrent VPN tunnels increases, the system may be rendered non-operational for specific high bandwidth applications such as video transmission.

5. Discussion

5.1. Centralized alternatives

P2PWNC was designed to operate in a fully decentralized manner. Here, we discuss the potential for a centralized alternative. We explore two cases. First, we assume a central entity with the sole responsibility of storing the system's history of transactions. This central receipt repository would alleviate the need for a gossiping protocol for receipt dissemination, relieving APs and local receipt repositories of the cryptographic overhead of verifying more receipts. On the other hand, such a scheme would still suffer from history pollution attacks such as the one described in Section 5.5. To make things worse, it could become a single point-of-failure. Also, in many settings (as is the case in most WCNs that do not rely on a central authority), it may be impossible to decide on a single trusted central receipt repository.

Second, relaxing the requirement that peers have free and uncertified identities, a PKI-based solution would be possible, with a trusted certification authority issuing team identifiers. This would protect the system from possible fake receipt DoS attacks (see Section 5.5) and *Sybil* [3] attacks, but would compromise its peer-to-peer nature, depriving it of its most attractive feature, the use of free IDs, and wasting the privacy enhancements that free IDs inherently offer. This might render the system inappropriate for many practical situations (e.g. WCNs) and hinder its organic growth. After all, proper design of P2PWNC reciprocity algorithms helps detecting even some sophisticated identity-related attacks [4].

5.2. Performance tradeoffs and choice of protocol parameters

Recalling that mobile users are expected to carry WLAN-enabled soft-phones that are battery powered (P2PWNC clients), while the AP software typically runs on top of embedded devices (such as the Linksys WRT54GS router) operating on AC power, we identify an interesting performance tradeoff. On the one hand, ECDSA implies smaller receipt sizes and fast and cheap (in terms of processing and, thus, battery usage) receipt generation, suitable for handheld devices. On the other hand, receipt verification is very slow and expensive, which, although not important in terms of battery utilization (receipt verifications are carried out on the AC powered Access Points), results in degraded throughput and, above all, fewer *sustainable*[4] concurrent P2PWNC sessions.

Let t_v denote the pure CPU time required for one receipt verification, T_{RREQ} denote the RREQ interval and n_{max} denote the number of concurrent P2PWNC sessions. Assuming that all sessions share the same digital signature scheme, an upper bound on the number of concurrent sustainable P2PWNC sessions is approximated by $n_{\max} \leq \lfloor T_{RREQ}/t_v \rfloor$. In practice, this number is always much smaller than T_{RREQ}/t_v, since T_{RREQ} represents wall clock time, while t_v represents pure CPU time and other CPU tasks are not considered in the formula. Therefore, given the above formula, a hotspot owner can choose between more accurate accounting and more sustainable P2PWNC sessions by modifying the T_{RREQ} value. A high T_{RREQ} means that much of the traffic generated in a session can be left unacknowledged (a session ends implicitly when a client fails to reply to the last RREQ) but more sessions can be handled, since receipt verification load is reduced (less frequent verification requests).

5.3. Roaming privacy

P2PWNC receipts are the only proofs that a transaction has taken place. As described in Section 2.1, a digital receipt includes the identifier of the providing team (that is the team's public key), but it does not disclose any information about the actual AP where service provision has taken place. Therefore, the roamer's location privacy is enhanced.

5.4. End-to-end security provision

Providing end-to-end security in a peer-to-peer manner in the multimedia communications architecture that we have proposed remains an open issue. Although the two endpoints of the multimedia call have set up secure VPN connections with their home gateways, the path between the two gateways is still unsecured.

If we suppose that the IP address of the home gateway of the caller has been communicated to the callee via a GSM SMS (see Section 3), it is straightforward for the GSM operator to sniff on the multimedia call, performing a simple man-in-the-middle (MITM) attack: it can modify the contents of the SMS pointing to a gateway of their own. The callee responds with the multimedia stream that comes unsecured from the VPN gateway of the callee to the gateway that belongs to the GSM operator. Then, the operator forwards the traffic to the caller and none of the call endpoints is aware that their traffic is being sniffed.

To combat such an attack and achieve true secure and private peer-to-peer multimedia communication, the unprotected part of the call has to be secured. A simple means is that of conveying the caller's public key to the callee during the call setup phase (GSM SMS exchange), and using it to exchange a shared key for traffic encryption. However, the callee has to verify that the public key of the caller has not been changed by an adversary (e.g. GSM operator) performing a MITM attack. Thus, after communication has been set up and assuming a voice or video call, the two parties can use some form of voice acknowledgement to verify that the public key exchanged is the appropriate. This way, an end-to-end secure VoIP or video call can be set up without resorting to trusted certification authorities.

[4] We define that a P2PWNC session is *sustainable* when the AP is capable of performing the receipt verification before the time when the next receipt request is scheduled to take place. More concurrent sessions, obviously, result in higher user-perceived verification times.

5.5. Denial-of-service attacks

Here, we only consider attacks that are directly related to the P2PWNC mechanism and not generic physical layer or network layer attacks. The most important one is implied by the operation of the gossiping protocol (Section 2.2). A malicious attacker may present the AP with a number of cryptographically sound but, otherwise, fake receipts[5]. Thus, CPU time will be spent on fake receipt verifications. More important, though, is the fact that, since receipt repositories have finite size and older receipts are being discarded and replaced by fresh ones (when the repository is full), the attacker may cause valid receipts to be replaced by dummy ones, thus polluting the provider's transactions history. However, this attack suggests pure malice on behalf of the attacker since he has no direct gain out of it.

Also, at the lower layers, a prospective contributor may perform physical layer jamming against the APs of his neighbors, so that only his APs can be detected by roaming peers, who will eventually request service only from him (and he will be rewarded with the service receipts). However, this attack can easily be detected.

6. Conclusion

We presented the design of a decentralized scheme for the provision of security- and privacy- enhanced ubiquitous Internet multimedia services in metropolitan areas. We discussed security related design and operation issues and evaluated aspects of its performance on low-cost resource-constrained devices that have become typical home WLAN equipment. Based on the above, we believe that our vision for a *secure* low-cost substitute to 2G/3G services is not far from its realization, especially in metropolitan areas.

7. Acknowledgement

This research is funded in part by the Greek GSRT under a project titled "E2E-Security over heterogeneous mobile/wireless networks."

8. References

[1] Athens Wireless Metropolitan Network, http://www.awmn.net

[2] N. Ben Salem, J.-P. Hubaux, and M. Jakobsson, "Reputation-based Wi-Fi deployment," *Mobile Computing and Communications Review* (MC2R), July 2005.

[3] J. Douceur, "The Sybil attack," In Proc. 1st International Workshop on Peer-to-Peer Systems (IPTPS), Cambridge, MA, March 2002.

[4] E.C. Efstathiou, P.A. Frangoudis, and G.C. Polyzos, "Stimulating Participation in Wireless Community Networks," In Proc. IEEE INFOCOM 2006, Barcelona, Spain, April 2006.

[5] FON, http://en.fon.com

[6] P.A. Frangoudis, "The Peer-to-Peer Wireless Network Confederation Protocol: Design Specification and Performance Analysis," M.Sc. Thesis, AUEB, 2005. Available as: http:// mm.aueb.gr/technicalreports/2005-MMLAB-TR-02.pdf.

[7] A.V. Goldberg and R.E. Tarjan, "A new approach to the maximum-flow problem," *Journal of the ACM*, vol. 35, no. 4, pp. 921-940, 1988.

[8] N. Koblitz, A. Menezes and S. Vanstone, "The State of Elliptic Curve Cryptography," *Designs, Codes and Cryptography,* vol. 19, pp. 173-193, 2000.

[9] Linksys Wireless-G broadband router, http://www.linksys.com

[10] Linspot, http://www.linspot.com/businessmodel.html

[11] S. Miltchev , S. Ioannidis , A. D. Keromytis, "A Study of the Relative Costs of Network Security Protocols," In Proc. USENIX 2002 Annual Technical Conference, p.41-48, June 10-15, 2002.

[12] NYCwireless, http://www.nycwireless.net

[13] The Openswan project, http://www.openswan.org

[14] OpenWRT Linux Distribution, http://openwrt.org

[15] The P2PWNC project website, http://mm.aueb.gr/research/P2PWNC/

[16] B. Patel, B. Aboba, W. Dixon, G. Zorn, and S. Booth, "Securing L2TP using IPsec," RFC 3193, November 2001.

[17] QTEK 9100 Pocket PC Phone ed., WLAN-enabled, http://www.qtek.nu/europe/products/9100.aspx

[18] Seattle Wireless, http://www.seattlewireless.net

[19] Speakeasy WiFi NetShare Service, http://www.speakeasy.net/netshare/

[20] Standards for Efficient Cryptography Group, "SEC1: Elliptic Curve Cryptography," September 2000. Available at http://www.secg.org.

[21] Standards for Efficient Cryptography Group, "SEC2: Recommended Elliptic Curve Domain Parameters," September 2000. Available at http://www.secg.org.

[22] É. Tardos and K.D. Wayne, "Simple Generalized Maximum Flow Algorithms," In Proc. 6th International Conference on Integer Programming and Combinatorial Optimization, pp. 310-324, 1998.

[23] The ttcp tool, http://ftp.arl.mil/ftp/pub/ttcp/

[24] Wireless Philadelphia Executive Committee, http://www.phila.gov/wireless

[25] C. Xenakis, N. Laoutaris, L. Merakos, I. Stavrakakis, "A Generic Characterization of the Overhead Imposed by IPsec and Associated Cryptographic Algorithms," *Computer Networks*, (to appear).

[5] Fake receipts: receipts that do not represent a real transaction and/or involve peers that do not exist. It is trivial to construct such receipts, since peer identities are *free*.