Design, Implementation, and Performance Analysis of DiscoSec – Service Pack for Securing WLANs

Ivan Martinovic, Paul Pichota, Matthias Wilhelm,
Frank A. Zdarsky, and Jens B. Schmitt

disco | Distributed Computer Systems Lab
University of Kaiserslautern, Germany
{martinovic, p_pichot, m_wilhel, zdarsky, jschmitt}@informatik.uni-kl.de

Abstract

To improve the already tarnished reputation of WLAN security, the new IEEE 802.11i security standard provides means for an enhanced user authentication and strong data confidentiality. However, the standard focuses on securing higher-layer data, i.e., protecting IEEE 802.11 data frames. The management frames used for connection administration are left unprotected and a wide spectrum of known attacks is still applicable and even extended against the IEEE 802.11i/IEEE 802.1X protocol execution.

This work describes DiscoSec, a service pack for "patching" WLANs against the most prominent vulnerabilities resulting in resource-depletion and impersonation attacks. DiscoSec provides DoS-resilient key exchange, an efficient frame authentication, and a performance-oriented implementation. By means of extensive real-world measurements the performance of DiscoSec is evaluated showing that even on very resource-limited devices the throughput is decreased by only 22 % compared to the throughput without any authentication, and by 6 % on more modern hardware. To demonstrate its effectiveness, DiscoSec is available as an open-source WLAN device driver.

1. Introduction

The various confidentiality and integrity vulnerabilities of Wired Equivalent Privacy (WEP) and the simplicity of mounting impersonation attacks by manipulating the sender's MAC address caused the bad reputation of IEEE 802.11 security. To regain back trust in this widespread technology the IEEE Task Group *i* ratification process successfully finalized the new security standard *802.11i*. The new standard provides a security framework composed of several known and approved protocols to ensure robust protection of wireless communication. An enhanced user authentication, a new underlying cipher, and reliable integrity verification finally enabled the protection of data equivalent to the security in wired networks.

However, IEEE 802.11i [1] focuses only on securing the user's data, i.e., it provides security for the *data frames* used to transport higher layer protocols, leaving the *management frames* used for channel and connection administration without any protection. The reason seems to be twofold. First, the tragic end of WEP left wireless clients without standardized protection giving rise to dispersion of proprietary solutions, hence the interoperability certification program (e.g., Wi-Fi Protected Access (WPA)) was impatiently awaited by both, the industry and the users. Secondly, management frames impact the *availability* of the IEEE 802.11 network, that especially in wireless networks, is the most vulnerable among all security goals. Due to the frequency jamming which is an indigenous property of wireless communication the importance of providing availability protection at the link-layer is often downgraded. Nevertheless, there is a significant difference between physical layer attacks aiming at the channel capacity, thus denying *any* communication, and link-layer attacks affecting the *services* provided by an access point and the connection states of wireless stations.

In its infrastructure mode an access point (AP) is controlling the wireless channel and providing authentication and association procedures to wireless clients. The importance of flawless operation, its availability to manage client associations, and administration of the wireless traffic directly impacts the overall user's security. For example, the execution of the IEEE 802.11i security standard is only possible if a wireless client reaches the final *authenticated and associated* connection state. State transitions within the IEEE 802.11 state machine are utilized by management frames, and by manipulating them even the sophisticated protection given by IEEE 802.11i is easily obviated (for attacks on IEEE 802.11i exploiting these vulnerabilities see [6]).

Furthermore, the typical Man-In-The-Middle (MITM) attacks in wireless networks are based on abusing unauthenticated management frames. After installing a rogue AP with a stronger signal strength, an adversary can simply change its MAC address to any of the already associated clients and, by sending an impersonated deauthentication or disassociation frame, it can bring a wireless client to its initial state [3]. Consequently, a wireless client is not able to transmit any data frames and must re-initiate the network discovery procedure which by default chooses the AP with a stronger signal, hence associating with the rogue AP. Although well known, these attacks are still applicable and various tools are available to facilitate their execution (e.g., [8] demonstrates wireless phishing within public hotspots).

To sustain the fast deployment of IEEE 802.11 technology there is a need for protection against such attacks. This work describes and evaluates DiscoSec, a solution against the most prominent vulnerabilities resulting in resource-depletion and impersonation attacks. A similar goal was also set within the IEEE 802.11 Task Group *w* which still remains in its early proposal stages. We therefore implement the concept of DiscoSec as an open-source IEEE 802.11 device driver to serve as a benchmark and prototype for future developments.

In Section 2, we discuss various security objectives influencing the design of DiscoSec. The key exchange, and implementation-related decisions are presented in Section 3, while Section 4 shows the impact of resource-depletion attacks and introduces a protection to mitigate them. In Section 5, we evaluate DiscoSec using three different hardware platforms and conduct real-world measurements to analyse the key exchange, frame authentication, and the impact of DiscoSec on the network throughput. Various design and implementation decisions were based on measurements using modern equipment and in this paper we therefore describe not only the final results, but also different lessons we have learned during our research.

2. Design Goals of DiscoSec

In contrast to wired networks where end-devices consist of comparable hardware capabilities and executing expensive computations do not present a performance problem, introducing cryptography-based protection in wireless networks opens various performance-related issues. Especially critical are protections utilizing stateful protocol executions and complex message exchanges which often result in new resource-depletion attacks. Such examples can also be found in IEEE 802.11i where a resource demanding protocol and unauthenticated exchange of key material were prone to various protocol-blocking attacks (e.g., [5, 6]). To mitigate such attacks and to allow interoperability between stations not supporting DiscoSec, we identified three most important requirements on which the solution should be based:

1. *Simple and lightweight authentication protocol*
2. *DoS resilient protocol execution*
3. *No alterations to current IEEE 802.11 state machine*

Simple and lightweight authentication is necessary for several reasons. Simplicity is a property affecting not only protocol design but also its implementation. In wireless communication where no assumption on a reliable channel should be made, protocols consisting of many round-trips (e.g., many message exchanges) often create deadlock vulnerabilities. The simplicity of authentication also assists us in reusing well-established cryptographic primitives available within the standard Linux (kernel) crypto API and the OpenSSL library, thereby minimizing the potential for faulty implementation.

The *lightweight* property of an authentication protocol focuses on the key exchange phase where the public key cryptography is used. To avoid many message round-trips we abandon the negotiation of security properties and rather utilize anonymous Diffie-Hellman (DH) key exchange. The idea behind this decision is to shift the key exchange phase to the very beginning of the communication where no resource reservation is made before the key exchange is finalized. At this stage no user identities are known but only their link-layer addresses, and therefore DiscoSec *binds* the sender's and receiver's link-layer address for the remainder of the session. It protects optional identity authentication within the later stages of communication utilizing more heavy-weight protocols. As a result, the key exchange is executed within only *one round-trip* (i.e. two messages) whilst supporting the second important design property - DoS resilient protocol execution.

DoS resilience of DiscoSec concerns both computational- and memory-depletion attacks. The key exchange is the most vulnerable part of the authentication protocol and its arbitrary initiation should be avoided. For this reason we implement a rate limitation of key exchange requests which takes advantage of broadcast communication to support the fair chance of associations among all stations. DoS Protection is provided as a configuration parameter and its dimensioning can be adapted to the performance characteristics of a dedicated AP.

To support legacy and DiscoSec protected stations within the same basic service set (BSS), we design and implement DiscoSec *without changing the current IEEE 802.11 state machine*. All required information is embedded within the existing frames as *Information Elements* (IEs), the key-value data structure reserved by the IEEE 802.11 standard for transmission of custom data. The authentication data is only processed if DiscoSec is implemented, otherwise it

	Management Frames	Data Frames
State 1	Beacon, Probe Req. /Resp., Traffic Indication Message, **Authentication Req./Resp., Deauthentication.**	None (infrastructure BSS)
State 2	**Association Req./Resp., Reassociation Req./Resp., Disassociation.**	None
State 3	**Deauthentication**	**All frames**

Table 1. IEEE 802.11 frame types and connection states within they are allowed to be transmitted. Bold frames are authenticated by DiscoSec.

is simply discarded by the legacy driver. The frame structure remains unchanged and for legacy stations DiscoSec enhanced AP has no impact on the association procedure.

Various other properties identified as performance vs. security tradeoffs are offered as configuration parameters of DiscoSec's implementation and left to a user.

2.1. Contribution

The security goals which DiscoSec fulfills are the *authentication* and *integrity* of management and (optionally) data frames exchanged between a wireless station and an AP. This eliminates the most prominent attacks based on sending fake or impersonated frames, such as Deauthentication and Disassociation attacks.

Table 1 provides an overview of all management and data frames used within the three connection states of wireless clients (authenticated frames are depicted as bold). Using DiscoSec, both station and AP hold a shared secret *before* entering the state two which is the first state demanding reservation of the AP's memory resources. Both participants are able to prove that all subsequent unicast frames are sent from the devices that participated in the network association procedure. Furthermore, due to its high performance DiscoSec offers authentication of all *data frames* transmitted during the session, and thus protects the execution of more heavy-weight protocols transmitted within them (e.g., IEEE 802.1X/EAPOL, IEEE 802.11i).

As a proof of concept DiscoSec is available as a ready-to-use open source WLAN device driver for all Atheros-based chip-sets.

3. DiscoSec's Design and Implementation

The design of DiscoSec followed the requirements discussed in the previous section. The cryptographic primitives used as building blocks for implementation of DiscoSec were based on their performance properties. Decisions such as the choice of the underlying cipher, using a block cipher-based message authentication code (CMAC), and elliptic curve cryptography (ECC) for key exchange were based on extensive measurements on dedicated APs (a more detailed discussion is given in Section 5).

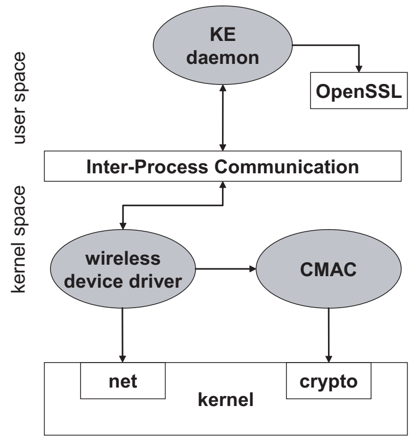

Figure 1. Architecture of DiscoSec

3.1. Architecture of DiscoSec

The architecture of DiscoSec is depicted in Figure 1. The overall functionality is split into modules and logically divided into three functional units: *(i)* the wireless LAN device driver that controls the WLAN hardware and contains the 802.11 network stack, *(ii)* the Key Exchange daemon which provides the public key cryptography utilizing primitives offered by the OpenSSL library. It processes the key exchange requests issued by the wireless device driver through Inter-Process Communication, and *(iii)* the CMAC kernel module which provides functions for calculating Message Authentication Codes (MACs) using the kernel's standard crypto API.

The calculation of the MACs is a time-critical operation, thus being implemented in *kernel space*. Contrary to CMAC, the Key Exchange daemon runs in *user space* with a lower priority. In case of a high CPU load, the Key Exchange daemon is scheduled less often, so the already associated stations are not influenced by expensive computations and their data throughput remains stable.

3.2. Terminology and Cryptographic Primitives Used

Table 2 shows the notation used in DiscoSec's frame authentication. All exchanged variables are defined as custom Information Elements appended to existing IEEE 802.11 frames.

The key exchange utilizes Elliptic Curve Diffie-Hellman (ECDH) as it enjoys the advantage of much smaller key

AP	Link-layer address of access point
STA	Link-layer address of wireless station
EC_{Param}	Elliptic curve parameters
PK_{AP}	Access point's public-key
PK_{STA}	Station's public-key
MK	Master key computed from ECDH
SK	Session key used for authentication
$MAC_{SK}(m)$	Message authentication code
AT	Association Token

Table 2. Notation used.

sizes compared to Diffie-Hellman based on the discrete logarithm problem. The public keys PK_{AP} and PK_{STA} are available in 128 bit length (expandable to 256 bit). EC_{Param} defines an elliptic curve over a finite field supported by the OpenSSL library and changeable through DiscoSec's configuration parameters.

The shared secret MK is computed from the ECDH key exchange using PK_{AP}, PK_{STA} and the station's and AP's private keys. It serves as a *master key* to derive key material for authentication.

The association token AT is used as *nonce* for computing a fresh session key SK for frame authentication, and additionally as DoS protection to control the rate of association requests (more information on associating tokens and DoS protection is given in Subsection 4.1).

The MAC_{SK} is a cipher-based message authentication code utilizing AES. We selected AES as an underlying cipher due to its availability within the IEEE 802.11i standard and good performance characteristics, it is provided within the Linux Crypto API (ver. 2.4+). The secure CMAC based on AES for authentication of messages with variable length was during the development of DiscoSec not available. Therefore we implemented RFC 4493 [13] which defines AES-CMAC and serves as a NIST recommendation for CMAC message authentication using AES block cipher [11].

3.3. Association Procedure - Key Derivation

We omit the detailed description of a public/private key initiation and a ECDH shared secret computation. Both methods are standardized and their implementations are given by OpenSSL ver. 0.9.8+ [7, 12]. In the following we describe DiscoSec specific parameter exchange and the derivation of the authentication key.

The key exchange is accomplished within a single roundtrip:

$$\text{AP} \rightarrow \text{STA} \;:\; PK_{AP}, EC_{Param}, AT$$

$$\text{AP} \leftarrow \text{STA} \;:\; PK_{STA}, AT$$

During start-up the AP initializes its key pair based on the elliptic curve parameters EC_{Param} (AP's key pair can also be precomputed and loaded during the start-up). The resulting PK_{AP} and EC_{Param}, are sent to the stations via periodically emitted Beacon frames or a triggered Probe Response frame depending on either active or passive network discovery.

The wireless station extracts the supplied values, generates its key pair based on EC_{Param} and computes the master key MK using *ECDH* method. The session key SK is created by applying the cryptographic hash function *SHA-256* on the MK and a previously received association token AT. The 128 least-significant bits of the hash are selected to provide the authentication key.

The reason for deriving the authentication key from the master key is to support a *key-caching* technique similar to the IEEE 802.11i standard. If the same wireless station decides to associate with the same AP and uses the static key pair, the computationally expensive *ECDH* key exchange can be omitted. The fresh SK is then derived by applying a single hash computation on the new association token and the cached MK.

The PK_{STA} and the AT are returned within the Authentication Request to the AP which computes the MK and SK analogously to the station side. This finalizes the key exchange and both participants use their session key SK as secret key for AES frame authentication ($MAC_{SK}(m)$).

This is also the most critical part of the association procedure. While the AP's key pair can be calculated offline and loaded during start-up, the session key derivation is triggered by an Authentication Request and its computation depends on PK_{STA} and AT. This opens a new vulnerability because the Authentication Request can easily be faked and the validation is only possible after the AP has derived MK by performing a complex modular computations. If the Authentication Request frames are received faster than the AP's transmission queue is processed, the AP can suffer from high frame loss and various operational anomalies. To prevent this kind of resource-depletion attack, DiscoSec provides a countermeasure based on an association rate control which is described in Subsection 4.

3.4. Frame Authentication - Variable vs. Fixed Frame Length

The challenge of frame authentication lies in its performance. While the low number of transmitted management frames only marginally increases computational load, the authentication of every data frame significantly stresses performance-limited APs. Consequently, data frame authentication directly impacts the throughput of the wireless connection.

For efficient frame authentication the most influential parameter is the frame's *length*. The expected Ethernet frame

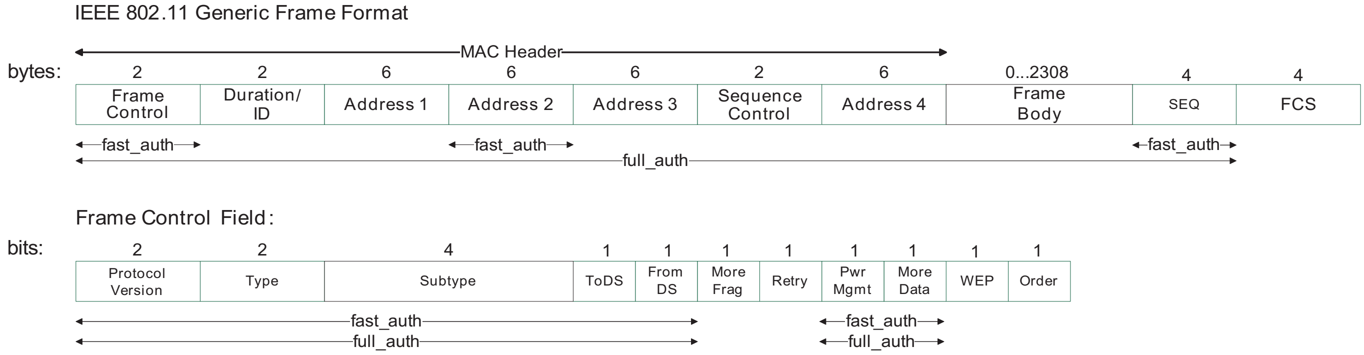

Figure 2. Authentication Modes: authenticated frame fields in $full\_auth$ and $full\_auth$.

length of max. 1500 byte is overrun in IEEE 802.11 networks and presently, frames up to 3000 byte are transmitted over the wireless channel. The reason lies in the proprietary features of various WLAN cards whose purpose is increasing throughput. For example the *SuperG [2]* extensions of Atheros utilizes so-called *FastFrame,* and *Bursting* techniques. FastFrame exploits the wireless channel more efficiently by increasing the amount of information contained in a single frame while Bursting increases throughput by sending multiple frames at once (without pausing between them). Although not standardized, these properties are common to various IEEE 802.11 vendors (under different names such as *SuperG*, *TurboG*, *Plus*) and their standardization should be finalized within the *802.11n* standard.

The authentication of such frames can often present a computational burden for performance-limited APs. To be able to support these extensions DiscoSec provides two modes of authentication - the $full\_auth$ mode for the APs with sufficient computational capabilities and the $fast\_auth$ mode for APs where full authentication of frames would result in a new performance bottleneck.

The *full_auth* mode is based on CMAC-AES and supports authentication of frames with variable lengths. In contrast, the $fast\_auth$ mode limits the data included in the MAC to a fixed size of 128 bits. In this mode only certain header fields are authenticated and the frame's payload is omitted from computation (both modes and authenticated frame fields are depicted in Figure 2). The authenticated fraction matches the block size of the AES cipher and can be authenticated within a single AES computation (the *AT* of 4 byte is included into computation, although not depicted in the figure). Accordingly, the authentication is more lightweight and may be performed much faster (for a quantitative comparison see section 5). On the other side, this presents a tradeoff between security and performance, i.e., complete authentication vs. higher throughput. While in our opinion the meaningful on-the-fly manipulation of the single bits during wireless transmission is hard to achieve and therefore *fast_auth* is sufficient for wireless communication, the decision on which mode to use is left to the user/AP as a configuration parameter.

3.5. Replay Protection

Frames transmitted over the wireless channel can easily be intercepted and used for replay attacks. To detect resending of such frames, DiscoSec implements replay protection by authenticating the frame's sequence numbers. The IEEE 802.11 generic frame format contains a sequence number field which it is 16 bits long out of which 12 bits are used for fragment count and only 4 bits as frame sequence number.

In order to provide a frame counter sufficient for long sessions and to avoid resynchronization problems, DiscoSec implements an independent 32 bit sequence number (SEQ) field. It is included in the MAC computation within both authentication modes (as shown in Figure 2).

The verification is based on accepting a received sequence numbers within a *window:*

$$seq_{previous} < seq_{current} < seq_{previous} + window + 1$$

This way, the false positive rejections of frames that are retransmitted due to loss or corruption are minimized. Frames containing a value less then the current sequence number are rejected. The magic number for the window length is usually selected around 10 which we also verified by real-world measurements (clearly, it depends on the wireless environment, hence it can be changed in DiscoSec's configuration).

4. Resource-depletion Protection

The association procedure in the infrastructure mode of an IEEE 802.11 network utilizes a stateful protocol execution which is prone to DoS attacks, especially to a memory-depletion attack of wireless access points (APs). After receiving an authentication request, an AP reserves memory

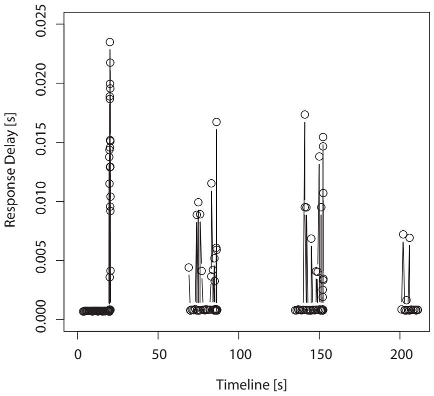

Figure 3. An outcome of resource-depletion attack - the unprotected AP periodically freezes.

for a client's connection state. Flooding an AP with a high number of fake authentication requests exhausts the AP's memory which consequently results in faulty operation or even a full device crash. As an example, the results of flooding a commonly used AP are shown in Figure 3. We flooded the AP with 100 authentication requests per second. Shortly after, the AP started freezing for longer periods of time, i.e., it would not respond to any frames. This example demonstrates how a simple attack heavily impacts the operation of an AP and it motivates the need for a resource protection.

Using cryptographic primitives on resource-limited devices, such an attack can even be extended to computational resources. In case of DiscoSec, flooding with authentication requests results in initiating many expensive computations and thus exhausting an AP's computational power. To avoid such attacks, DiscoSec provides simple but effective protection based on a rate limitation.

4.1. DiscoSec's DoS Protection

The protection is based on using *association tokens* which allow only a certain number of authentication requests per second.

The AP generates a set Θ_t containing association tokens which are 32 bit randomly chosen numbers. They are published at time t and applied using the following concurrent phases:

- *publication phase* P_t - the AP broadcast Θ_t to surrounding stations,
- *acceptance phase* A_t - the AP allows associations containing unused tokens from Θ_{t-1}.

Set Θ_t is *repeatedly* sent within the Beacon frames for the duration of the time interval $[t,\, t+1[$. The Beacon frame additionally contains the *Counter Field* which reports the number of Beacon frames until $t+1$.

The association procedure proceeds as follows: after receiving Beacon frame, the wireless station *randomly* chooses one token from Θ_t. It waits until the Beacon frame signals the beginning of $t+1$ and then sends the authentication frame containing the chosen token. If the token is unused the AP accepts the station's request and initiates the key exchange.

Using this mechanism successful authentication is independent of the time a station discovers tokens. Every station knows the beginning of A_{t+1} and possesses Θ_t therefore authentication success only depends on the CSMA/CA media access protocol equal to transmitting any frame using the IEEE 802.11 contention access mechanism. Choosing the random token helps legitimate STAs to increase their chance of successful authentication. To be certain that no legitimate station can authenticate, an attacker must be able to send all the tokens before the legitimate station's requests and it must succeed for each published Θ_t.

The implementation of this protection is simple as it only requires Θ_{t-1} and Θ_t to be saved at the AP. The length of the Counter Field is 1 byte and Beacons are per default broadcast every 100 ms. The number of tokens within Θ_t depends on the performance characteristics of the dedicated AP.

During our measurements the performance weakest AP could afford 10 authentications/s, i.e, every second the AP publishes 10 new tokens and accepts 10 tokens. It is important to mention that the tokens are only verified by the AP if the DoS Protection is enabled. On the other hand, running with DoS Protection, only the stations supporting the token mechanism can associate. This tradeoff is the unavoidable consequence of extending the AP's protection functionality.

While the primary objective to protect AP's resources and assure its operational stability is fulfilled within this version of DiscoSec, more sophisticated techniques to differentiate between legitimate and attacker's requests are part of our current research.

5. Performance Analysis of DiscoSec

5.1. Evaluated Platforms and Methodology

The selection of platforms for testing DiscoSec's performance focused on hardware discrepancies in order to represent the computational capabilities of broadly available devices. Their hardware characteristics are shown in Table 3.

The performance-weakest device is a 4G AccessCube[1]. The device is from the year 2004 and runs on an architecture other than x86, thus making cross compiling necessary.

[1] http://www.meshcube.org

Device	CPU [MHz]	RAM [MB]	Kernel
Cube	MIPS, 324	64	2.6.14
Routerboard	Geode, 266	256	2.6.17
Laptop	Pentium 3, 1400	1024	2.6.17

Table 3. Platforms Used.

The other two devices are a Routerboard[2] 230 using Voyage Linux[3] 0.3 and a medium-class Laptop operating in the master mode of the wireless device driver, i.e., offering authentication and association procedures.

To provide insight into all authentication-related delays DiscoSec was configured to protect both management and data frames. For throughput measurements we generated a continuous stream of UDP packets at various bit rates and under various AP utilizations. For measurements of key exchange and MAC computation we set the AP utilization to levels of 0 %, 15 %, 30 % and 50 % while monitoring delay as a response variable. The utilization was increased either by using already associated clients sending with the maximal throughput or artificially by additional CPU computations (if frame transmission did not result in a high AP utilization). The measurements were repeated 10 times and depicted results represent the mean with 0.95 confidence intervals.

Due to space limitation, the analysis of key exchange and frame authentication is given for the performance-weakest device AP_{Cube}, the final throughput results on overall DiscoSec performance are provided for all three devices.

5.2. Key Exchange and Frame Authentication

Before going into details of the delays introduced by key exchange, we briefly mention state-of-the art delays imposed by the IEEE 802.11i security standard. For mutual identity authentication the security standard requires an Authentication Server that undertakes the shared secret computation instead of the AP. In [9] measurements show that the IEEE 802.11i delay imposed by the key exchange using mutual authentication (e.g., EAP-TLS) varies between $\approx$ 300 ms and 4 s, depending on different platforms and various implementations of the standard.

Concerning DiscoSec's key exchange, Figure 4 depicts delays using different key sizes. The length of 128 bit elliptic curve key takes only $\approx$ 79 ms on the performance weakest device. The varying AP utilization does not influence the key exchange much and at 50% utilization the 128 bit key exchange remains under 400 ms. Clearly, longer keys increase computational time, nevertheless even the key exchange using 224 bit keys (equivalent to 2048 bit RSA public key) remains under 600 ms.

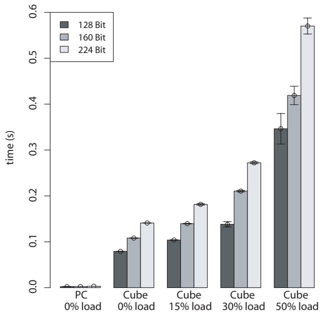

Figure 4. Cost of key exchange using ECDH under various AP loads and different key sizes.

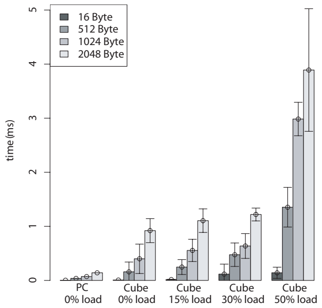

Figure 5. Computation of MAC for various message lengths.

In contrast to the shared secret which is generated only once at the beginning of the session, the MAC is calculated for each transmitted management and data frame, implicitly influencing the connection throughput.

In order to evaluate the measurement results given in Figure 5, it is important to consider the impact of MAC computation on the frame transmission. The maximum throughput of the AP_{cube} is around 29 Mbit/s using a plain driver without any extensions. This means that every $\approx 433\,\mu s$ a packet is transmitted. Viewed purely calculative, if the MAC generation takes just as long, which includes the overhead imposed from the driver, then the data rate will halve. On the tested hardware, processing of 1024 bytes of data already takes $\approx 400\,\mu s$, hence not leaving much space. Nevertheless, the same figure shows that the computation time remains stable and varies less (given by the interval length of the confidence intervals) for all keysizes if load is under 50 %, otherwise delay and its variance dramatically increase exhibiting the device's computational limits. For this reason, the *fast_auth* mode becomes inevitable. Using *fast_auth* the computation of authenticated data equals 16 bytes which requires AES key length of 128 bit and does

[2]http://www.routerboard.com
[3]http://linux.voyage.hk/

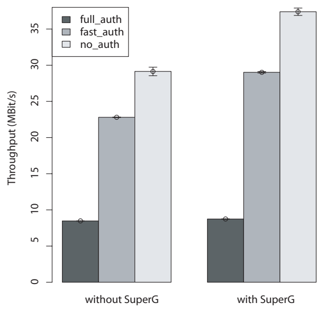

Figure 6. Throughput using standard transmission vs. SuperG features.

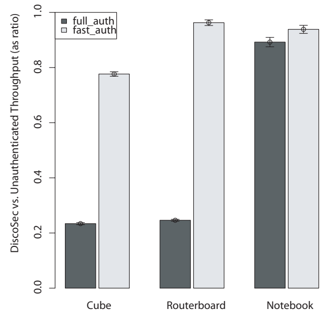

Figure 7. Throughput analysis of all tested platforms.

not exceed $\approx 150\,\mu s$ even at 50 % CPU load. This significantly relieves the computational burden resulting in much higher throughput as shown in the next subsection.

5.3. DiscoSec Featuring SuperG

The throughput comparison between plain and *SuperG* enhanced transmission is depicted in Figure 6 for three different configurations: *no_auth*, *fast_auth*, *full_auth*.

Considering the frame transmission without SuperG extensions, the *no_auth* bar denotes the maximum possible unauthenticated throughput of 29 Mbit/s equal to transmission without DiscoSec which serves as reference. Using *full_auth*, the complete IEEE 802.11 frame is authenticated. This security feature is the most computationally demanding and AP_{Cube} offers only 8 Mbit/s throughput.

Using the *fast_auth* mode the AP relaxes the computational requirements and achieves data rates of $\approx$ 23 Mbit/s (78% of unauthenticated throughput). This scenario shows the importance of providing *fast_auth* as a tradeoff parameter for performance-limited APs.

Enabling the SuperG extensions (FastFrame and Bursting) leads to shorter transmission delays and larger frames, increasing the *no_auth* throughput to 37 Mbit/s. But more importantly, since SuperG does not impact the IEEE 802.11 header, the computational effort of *fast_auth* mode is equal to a transmission without SuperG features although more data is being transmitted. Therefore, even on a very performance-limited device like AP_{Cube}, using SuperG with *fast_auth* authenticated transmission is at $\approx$ 29 Mbit/s (78%).

5.4. Overall Throughput

Until now, the presented analysis focused only on the weakest measured device. By using the *fast_auth* mode, performance degradation can be mitigated, though not eliminated. The trend of modern APs aims at offloading computations of cryptographic primitives, especially symmetric ciphers to specialized hardware. For example, the new generation of Geode CPUs features a hardware implementation of the 128 bit AES cipher and a true random number generator.

In our measurements we analyzed an older version of the Geode CPU within a *Routerboard RB200.* It is a multifunctional device running at only 266 MHz (less then AP_{Cube}), without any special-purpose hardware for faster computations. It uses a Gode x86 SC1100 processor, equivalent to the Intel Pentium MMX architecture. Although its low CPU clock frequency does not allow for much faster computation, in *fast_auth* it achieves 97 % of the possible throughput. The modest-looking 2-3% throughput increase of *full_auth* mode compared with AP_{Cube} implies that Routerboard succeeds in authenticating $\approx$ 1 Mbit/s more data. Using hardware capabilities of an older Notebook running as an AP, it shows exemplary authentication performance. The throughput of both, *full_auth* and *fast_auth* mode is at 89 % and 94 %, respectively.

To summarize, this section provided an overview of what to expect of the network throughput using computationally-limited devices. It demonstrates the importance of offering security vs. performance tradeoffs which in turn may smoothen throughput differences among heterogeneous hardware platforms.

6. Related Work

Concerning attacks based on unauthenticated management and data frames [3] demonstrates their devastating effect on IEEE 802.11 networks. Based on the same vulnerabilities, in [6] various attacks are successfully mounted even against the new security standard IEEE 802.11i. While the empirical demonstration is a frequently used method to illustrate the problem of link-layer security, protection against such attacks prevalently remains conceptual. Only [4] discusses the implementation issues of a proposed solution. The authors employ two protocols, SIAP and SLAP, to establish a secure association utilizing public key infrastructure. While their solution offers encryption, it also modifies the IEEE 802.11 state machine and requires a SIAP server.

In commercial products, Cisco offers a feature called Management Frame Protection (MFP), but there is regrettably no detailed information other than white papers [14]. Interestingly, MFP does not seem to be a client-side supported feature, and thus only protects APs, while clients remain vulnerable to management frame attacks.

The inital idea of the authentication mechanism used in DiscoSec was described in [10]. In this paper we significantly improved and implemented the concept, and analysed its performance. To the best of our knowledge, DiscoSec is the first solution with a design supporting the IEEE 802.11 state machine, extensively tested on performance-limited hardware and available[4] for use on present devices.

7. Conclusion

This work described DiscoSec, a lightweight authentication protocol designed to protect WLANs against most prominent attacks based on resource-depletion and impersonation of management and data frames. DiscoSec followed the idea of "patching", i.e., providing a small, effective and easily applicable solution to a variety of devices.

During development of DiscoSec, we came across various design and implementation decisions such as providing DoS-resilient key exchange, efficient authentication, support for throughput-increasing features like SuperG, and usage of widely-accepted cryptographic primitives. Most of these decisions are offered as configuration parameters to facilitate balancing between security and performance tradeoffs. Using real-world measurements, we demonstrated that even a performance-limited device achieves 78 % of maximum throughput, while using a more powerful device the price paid is only 11 % and 9 % of the throughput decrease for full and fast authentication, respectively.

8. Acknowledgement

We gratefully acknowledge the *madwifi.org* project which was used as basis for implementing DiscoSec.

References

[1] IEEE 802.11i/D10.0. Security Enhancements, Amendment 6 to IEEE Standard for Information Technology. IEEE Standard, April 2004.

[2] Atheros. SuperG – Maximizing Wireless Performance . Available at www.super-g.com, (last accessed 12.10.2007).

[3] J. Bellardo and S. Savage. 802.11 Denial-of-Service attacks: Real Vulnerabilities and Practical Solutions. In *Proceedings of the USENIX Security Symposium*, pages 15–28, August 2003.

[4] D. Faria and D. Cheriton. DoS and Authentication in Wireless Public Access Networks. In *Proceedings of the 2004 ACM Workshop on Wireless Security*, pages 47–56, September 2002.

[5] C. He and J. C. Mitchell. Analysis of the 802.11i 4-way handshake. In *Proceedings of the 2004 ACM Workshop on Wireless Security*, pages 43–50, October 2004.

[6] C. He and J. C. Mitchell. Security Analysis and Improvements for IEEE 802.11i. In *Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS'05)*, pages 90–110, February 2005.

[7] IETF. Diffie-Hellman Key Agreement Method. RFC 2631, 1999.

[8] I. Martinovic, F. Zdarsky, A. Bachorek, C. Jung, and J. Schmitt. Phishing in the Wireless: Implementation and Analysis. In *Proceedings of the 22nd IFIP International Information Security Conference (SEC 2007)*. Springer, May 2007.

[9] I. Martinovic, F. Zdarsky, A. Bachorek, and J. Schmitt. Introduction of IEEE 802.11i and Measuring its Security vs. Performance Tradeoff. In *Proceedings of the 13th European Wireless Conference, France*, April 2007.

[10] I. Martinovic, F. A. Zdarsky, and J. B. Schmitt. On the Way to IEEE 802.11 DoS Resilience. In *Proceedings of the Workshop on Security and Privacy in Mobile and Wireless Networking (in conjunction with IFIP Networking 2006), Coimbra, Portugal*. Springer LNCS, May 2006.

[11] NIST Special Publication 800-38B. Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005.

[12] SECG. Elliptic Curve Cryptography, Standards for Efficient Cryptographic Group. Available at www.secg.org/collateral/sec2.pdf, (last accessed 08.10.2007).

[13] J. Song, R. Poovendran, J. Lee, and T. Iwata. The AES-CMAC Algorithm. RFC 4493 (Informational), June 2006.

[14] www.cisco.com; Document ID: 82196. Infrastructure Management Frame Protection (MFP) with WLC and LAP Configuration Example, (last accessed 28.05.2007).

[4] DiscoSec's source code for using it as a wireless device driver is available at http://disco.informatik.uni-kl.de/downloads/