DESIGN OF A PASSWORD-BASED AUTHENTICATION METHOD FOR WIRELESS NETWORKS

Andrea Manganaro, Mingyur Koblensky and Michele Loreti
Dipartimento di Sistemi e Informatica, Universita' di Firenze, Viale Morgagni 65 - 50134 Firenze, Italy

Keywords: EAP methods, Password-based authentication, SRP-6.

Abstract: In recent years, amendments to IEEE standards for wireless networks added support for authentication algorithms based on the Extensible Authentication Protocol (EAP). Available solutions generally use digital certificates or pre-shared keys but the management of the resulting implementations is complex or unlikely to be scalable. In this paper we present EAP-SRP-256, an authentication method proposal that relies on the SRP-6 protocol and provides a strong password-based authentication mechanism. It is intended to meet the IETF security and key management requirements for wireless networks.

1 INTRODUCTION

Securing communications over a wireless network requires protocols that provide both mutual authentication between the parties and correct integration with the available cryptographic algorithms. A common approach is that based on the Extensible Authentication Protocol (EAP) (Aboba et al., 2004), since it provides a generalized framework for the execution of arbitrary authentication mechanisms between two entities, a client and an authentication Server. Protocols that rely on EAP are usually called *methods*.

Amendments to IEEE standards 802.11 and 802.16 have defined the support for EAP in Wi-Fi and WiMax systems respectively. In both these environments the authentication methods are required to meet RFC 4017 (Stanley, 2005) that concerns security requirements, and the EAP Key Management Framework (Bernard Aboba, 2006).

EAP methods can be divided into four main categories, according to their authentication mechanism.

PKI-based methods rely on a Public Key Infrastructure (PKI) in order to manage the digital certificates exchanged between the parties. EAP-TLS (Aboba and Simon, 1999) is a well-known and widely implemented example of this kind of approach. Nevertheless PKI dependency often results in a considerable design complexity and digital certificates management is unlikely to be simple.

Tunneled methods provide an encrypted channel inside which two parties can perform a protected authentication procedure. Digital certificates are still required, but this is mandatory only on Server-side. PEAP (Palekar et al., 2004) and EAP-TTLS (Funk, 2005) fit into this category. These methods have the advantage of simplifying digital certificates management but they still depend on a PKI.

Pre-Shared-Key (PSK) methods perform authentication using cryptographic algorithms that rely on one or more secret keys shared between the parties without needing digital certificates. EAP-PSK (Bersani and Tschofenig, 2007) is believed to be the most representative for this category. PSK-based solutions could simplify network security management in certain evironments, unfortunately they have also some noticeable limitations:

- it is hard to implement key generation with an adequate entropy level;
- key distribuition and management could become complex and not scalable for large evironments;
- in general, such methods are not suitable to derive PSKs from users' passwords.

As a consequence, PSK methods are commonly used only in small environments.

Manganaro A., Koblensky M. and Loreti M. (2007).
DESIGN OF A PASSWORD-BASED AUTHENTICATION METHOD FOR WIRELESS NETWORKS.
In *Proceedings of the Second International Conference on Wireless Information Networks and Systems*, pages 9-16
DOI: 10.5220/0002147200090016

Password-based methods typically are protocols capable of performing authentication using only users'passwords as credentials. Such solutions are attractive, since passwords are by far the most widespread credential type. Nevertheless, to the best of our knowledge, no password-based method has been yet standardized or recognized as "secure".

1.1 Applicability

In production wireless environments, the majority of the adopted EAP methods are PKI-dependent: PEAP and EAP-TLS are the most implemented methods today. As mentioned above, such solutions tend to be expensive since PKI design and management are pretty complex. It is not trivial to implement a X.509 infrastructure where certificates are generated, distributed and revoked efficiently. Handling of certificates lifecycle is a very delicate aspect for every PKI and any design error could result in a security weakness. Moreover, clients should be able to check the authenticity of certificates. To do that in practice, clients use the CA public key with the proper verification algorithm (RSA or DSS). As noted in (Skoudis, 2002), this may lead to a *weak trust-model*, since it is strongly based on client's system settings and user's choices.

There is a substantial lack of real alternatives to PKI-dependent solutions, consequently the authors believe this is a relevant research topic to develop. Having a robust authentication method that does not use digital certificates would be desiderable, since it could offer a less expensive solution for wireless communication systems.

In this paper we present EAP-SRP-256, a new EAP method proposal that allows password-based authentication. EAP-SRP-256 is based on the SRP-6 protocol and it has been designed to be suitable for both Wi-Fi and WiMax networks.

2 THE SRP-6 PROTOCOL

The Secure Remote Password (SRP) protocol (Wu, 1997) is a password-based authenticated key-exchange protocol designed to resist both active and passive attacks. The SRP-6 protocol is an earlier improved version of SRP that has been defined in (Wu, 2002) and included within the IEEE-P1363.2 standardization process. In this section we give a brief overview on protocol properties and functionalities.

2.1 Foundation

From the mathematical standpoint SRP-6 is a *Diffie-Hellman Key Exchange* (Diffie and Hellman, 1976) variant that relies on discrete logarithm properties and uses the parameters shown in Table 1.

Table 1: Mathematical notation for SRP.

I	client's identity
n, g	group parameters (prime and generator)
k	constant value derived from n and g
s	salt (random value)
P	client's password
x	a private key derived from I, P and s
v	client's password verifier
A, a	client's public and private values
B, b	server's public and private values
H	a one-way hash function
K	the SRP session key

2.1.1 Modular Exponentiation

All exponential computations are performed in a finite field GF(n). In other words, given a large prime n, all operations are performed modulo n. Value g is required to be a primitive root modulo n and it is called *generator* in a GF(n). For the generation of this group parameters, existing implementations of SRP commonly use a predefined set of values that meets the required constraints. See e.g. (Taylor et al., 2006).

2.1.2 Hash Functions

SRP-6 requires a one-way hash function H. In real world implementations the SHA-1 or SHA-256 algorithms are commonly believed adequate, since into this context they are not susceptible to the currently known attacks to hash functions reported in (Hoffman and Schneier, 2005).

2.1.3 Parameters Initialization

Each client must be prior accounted by an authentication Server using client's identity (I) and password (P). Every account is created on server-side first generating s, a pseudorandom value called *salt*, and then computing the following parameters:

$$x = H(s, I, P) \tag{1}$$

$$v = g^x \bmod n \tag{2}$$

$$k = H(n,g) \tag{3}$$

Thereafter the authentication Server stores the triplet (I,s,v) related to the accounted client with identity I and n,g group parameters. Server does not store the P value in any way.

2.2 Protocol Overview

The SRP-6 protocol performs mutual authentication between a client and an authentication Server, deriving a different session key (K) at the end of each successful authentication process. The length of K depends on the chosen hash function properties.

Figure 2 shows the protocol flow that occurs between the parties. Note that messages are sent "in clear" and exponential operations must be considered modulo n.

Table 2: The SRP-6 protocol.

Client		Server
	$\xrightarrow{I}$	lookup (I,s,v)
$x = H(s,I,P)$	$\xleftarrow{n,g,s}$	
$A = g^a$	$\xrightarrow{A}$	$B = kv + g^b$
$u = H(A,B)$	$\xleftarrow{B}$	$u = H(A,B)$
$S = (B - kg^x)^{a+ux}$		$S = (Av^u)^b$
$M_1 = H(A,B,S)$	$\xrightarrow{M_1}$	verify M_1
verify M_2	$\xleftarrow{M_2}$	$M_2 = H(A,M_1,S)$
$K = H(S)$		$K = H(S)$

SRP-6 also allows message reordering to perform the authentication in a more efficient way, reducing the required protocol rounds. Optimized message ordering requires only two rounds and it is useful in practice for limiting the channel overhead. Message reordering is shown in Table 3.

Table 3: SRP-6 with optimized message ordering.

Client	Server	sent parameters
	$\Longrightarrow$	I
	$\Longleftarrow$	n,g,s,B
	$\Longrightarrow$	A,M_1
	$\Longleftarrow$	M_2

2.3 Security

The security analysis of any authentication protocol could not easily performed with formal methods. For instance, the *Dolev-Yao model* (Dolev and Yao, 1981) is applicable with certain constraints but it is not adequate to deal with primitives such as Diffie-Hellman exponentiation (Millen and Shmatikov, 2003). Moreover protocol security becomes undecidable with more general models (Heintze and Tygar, 1996), since the set of states to be considered is huge or infinite.

In recent years, there has been interest in proving the security of password-based protocols using the *ideal-cipher model* (Bellare et al., 2000) and there is evidence of its applicability (Bellare and Rogaway, 2000) even for SRP. Unfortunately it has been also demonstrated (Zhao et al., 2006) that a provable security in ideal-cipher model does not necessarily say that the instantiation of the protocol is secure. Consequently, the applicability of a security analysis with formal methods to password-based authentication protocols has not yet been proved.

Despite this formal limitation, the SRP-6 protocol could be considered *inherently secure*, since its mathematical structure can be reduced to the widely studied Diffie-Hellman problem (Diffie and Hellman, 1976). Consequently it is possible to prove its effective security against active and passive known attacks. See e.g. (Ferguson and Schneier, 2003).

3 DESIGN OF EAP-SRP-256

EAP-SRP-256 is a password-based authentication method that has been designed to operate in wireless networks and relies on the SRP-6 protocol. This EAP method allows mutual authentication between a client and an authentication Server, deriving two sets of symmetric keys during the protocol execution. Clients' passwords are the only needed credentials.

The development of EAP-SRP-256 is part of an academic research project[1]. The early design of the authentication method was given by (Manganaro, 2005) that has developed protocol specifications, design rationale and security analisys. A noticeable result was the open source implementation for Wi-Fi networks that has been recently presented by (Koblensky, 2006).

3.1 Architecture

The architecture of EAP-SRP-256 consists on seven building blocks that provide specific features and have

[1] http://rap.dsi.unifi.it/eap-srp-256/

one or more functional dependencies with the other blocks. It is shown in Figure 1.

The SRP-6 protocol gives the basic mechanisms for mutual authentication and primary session key establishment. The EAP method provides message-integrity protection (using the HMAC-SHA-256 algorithm) and data encryption for some data exchanged between the parties. Moreover, it uses a key derivation scheme that relies on the *Modified Counter Mode* (MCM), a block-cipher based function. The SHA-256 and AES-256 algorithms work as function primitives. EAP-SRP-256 optionally supports a Pseudo-random Number Generator (PRNG) compliant with the ANSI X9.31 and FIPS-140-2 standards.

Method specifications are completely based on the above mentioned architecture. They meet the security requirements defined in (Stanley, 2005) and the EAP Key Management Framework. Implementation details are discussed in section 3.3.

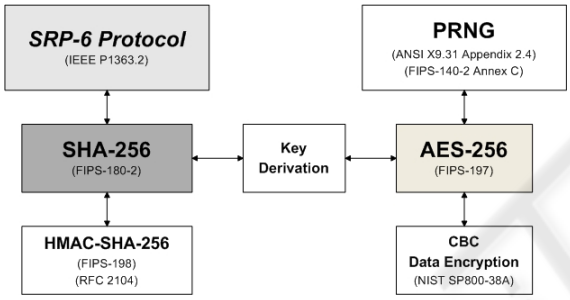

Figure 1: Architecture of EAP-SRP-256.

3.2 Method Overview

EAP-SRP-256 is a 4 round authentication method that encapsulates the optimized version of the SRP-6 protocol while adding some functionality, such as message integrity protection and a key derivation mechanism.

During protocol conversation the parties exchange encrypted data that contains pseudorandom values which are used for several purposes. At the end of a successful protocol execution, client and Server are mutually authenticated and own a set of symmetric session keys.

3.2.1 Message Flow

The entire message flow is shown in Table 4 and it consists of 9 messages that are formatted according to the EAP specifications: Server messages are considered *requests*, client messages are *responses*.

Table 4: The EAP-SRP-256 message flow.

Client	Server
	$EAP-Request\ Identity \longleftarrow$
$SRP-Anonymous \longrightarrow$	
	$MS,ServerName \longleftarrow$
$<id\_value>,MS,AS \longrightarrow$	
	$AS,N,g,s,B \longleftarrow$
$A,M1,h\_Data_1,h_1 \longrightarrow$	
	$M2,h\_Data_2,h_2 \longleftarrow$
$h\_Data_3,h_3 \longrightarrow$	
	$EAP-Success/Failure \longleftarrow$

In the case where a successful authentication occurs within EAP-SRP-256, the protocol conversation will appear as follows:

$$EAP-Request\ Identity \longleftarrow$$

Every EAP method is required to start with this explicit request. In IEEE 802.11 implementations this message is sent usually by the Access Point.

$$SRP-Anonymous \longrightarrow$$

This message is the conventional Response-Identity required by EAP. In this case it contains a default string (SRP-anonymous) that manifests the intention of the client to proceed with EAP-SRP-256 authentication without identifying himself explicitly.

$$MS,ServerName \longleftarrow$$

Server starts this specific EAP method, sending a nonce (MS) and optionally a string that represents Server name. Server is requiring client's identity.

$$<id\_value>,MS,AS \longrightarrow$$

Client sends a pseudonym ($< id\_value >$), the nonce received from Server (MS) and a new one (AS). The pseudonym identifies uniquely the client but does not correspond to its real username.

$$AS,N,g,s,B \longleftarrow$$

This message contains all the parameters needed for starting SRP-6 protocol, plus the AS nonce is sent back. After receiving this message, client performs SRP-6 computations and derives a set of ephemeral keys.

$$\xrightarrow{A,M1,h\_Data_1,h_1}$$

This message provides the "Secure Client Authentication" (SCA). Client sends its SRP-6 parameters, adding 64 bytes of encrypted data (h_Data_1) and h_1, a 32 byte *Keyed-Hash Message Authentication Code*, also know as HMAC. After receiving this message, Server performs SRP-6 computations and derives a set of ephemeral keys.

$$\xleftarrow{M2,h\_Data_2,h_2}$$

This message provides the "Secure Server Authentication" (SSA). Server completes SRP-6 protocol and sends 64 bytes of encrypted data (h_Data_2) and a 32byte HMAC (h_2). After receiving this message, the client derives a set of session keys.

$$\xrightarrow{h\_Data_3,h_3}$$

This message provides "Secure Method Confirmation" (SMC). Client sends 64 bytes of encrypted data (h_Data_3) and a 32 byte HMAC (h_3). With SMC, client confirms securely the correctness of encrypted data that parties have previously exchanged. After receiving this message, client derives a set of session keys.

$$\xleftarrow{EAP-Success/Failure}$$

Every EAP method is required to end with a *success* or *failure* message related to the authentication process.

3.2.2 Considerations

Initial messages merely start the authentication method while negotiating SRP-6 parameters. Note that in this case the pseudonym is used instead of the usual I value.

With the SCA, SSA and SMC messages, both parties perform mutual authentication and derive two different sets of symmetric keys referred as *ephemeral* and *session oriented*. Key derivation details are discussed in section 3.3.4.

The $h\_Data_{1,2,3}$ fields contain encrypted pseudo-random values, called *seeds* and *challenges*, that both the parties use for generating pseudonyms and session keys.

3.3 Implementation Details

3.3.1 Hash Function

The authentication method uses SHA-256 as hash function. The choice follows current best practices and the considerations included in (Hoffman and Schneier, 2005).

3.3.2 Cryptographic Primitive

EAP-SRP-256 uses the AES-256 algorithm (Daemen and Rijmen, 2002) as cryptographic primitive in order to:

- encrypt the $h\_Data_{1,2,3}$ parameters exchanged between the parties;
- run over the MCM-based key derivation mechanism;
- be used (optionally) for the PRNG engine.

The recommended mode of operation for encryption is the *CTR mode* as described in (Dworkin, 2001).

AES is a *de-facto* world standard and, since its introduction in 1999, there have been few cryptanalytic advances despite the efforts of many researchers (Dobbertin et al., 2004).

3.3.3 Message Integrity Protection

In order to avoid packet-modification attacks, EAP-SRP-256 uses authenticators ($h_{1,2,3}$) for each message that contains encrypted data. They are *Keyed-Hash Message Authentication Codes* (HMAC) (Krawczyk et al., 1997) that use the SHA-256 function and a 256 bit symmetric ephemeral key.

Authenticators are computed following the *Horton principle* (Wagner and Schneier, 1996), using a different key for each method execution.

3.3.4 Key Derivation

Key derivation is a crucial aspect for authentication and method security. EAP-SRP-256 derives two sets of symmetric keys in order to perform data encryption, HMAC computation and the exportation of keying material required by the EAP Key Management Framework.

The entire mechanism is based on the *Modified Counter Mode* (MCM), a block cipher expansion function that transforms a single input block into t blocks, where $t \geq 2$ and each output block is of the same length of the input. It has been demonstrated in (Gilbert, 2003) that, under certain constraints, the MCM resulting output is *secure* according the Luby-Rackoff paradigm (Luby and Rackoff, 1988), meaning that it will not be distinguishable from a perfect random function. In practice this results in two great properties:

- produced output cannot be guessed, even knowing the input;

- when the input is kept secret, it cannot be guessed, even if output security is broken.

EAP-SRP-256 applies AES-256 to the MCM, satisfying the required constraints and inheriting the properties above. Figure 2 shows the key derivation mechanism with all the involved components.

As previously mentioned, method execution produces two distinct sets ok keys. Ephemeral keys ($TEK_{1,2}$ and DK) have a lifetime limited to the method execution and they are mainly used for data encryption and HMACs computation.

Session oriented keys (MSK and $EMSK$) are those exported by the EAP method as keying material for the available crypto algorithms used at the OSI-layer 2. For implementations compliant with the IEEE 802.11i standard, this would correspond to the keying material used by RSN/WPA specifications for encryption.

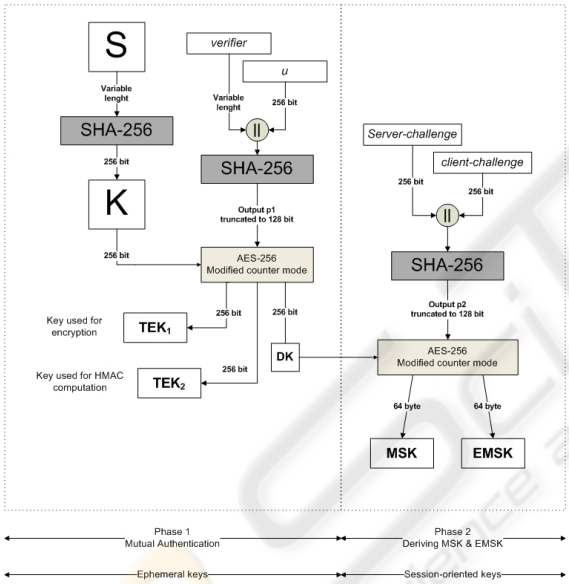

Figure 2: Key derivation for EAP-SRP-256: keys S and K are produced by the SRP-6 protocol execution.

3.3.5 Fragmentation

Since the parties exchange variable length parameters, a message fragmentation mechanism must be provided. To do so, EAP-SRP-256 supports fragmentation similarly to EAP-TLS, using a *fragment acknowledgment* scheme.

Fragmentation level primarily depends on the value of n, since it influences the dimensions of the biggest SRP paramenters, A and B. The value of n is chosen from a standard predefined set and its binary representation can grow up to 8192 bits. Nevertheless, protocol overhead determined by this feature is minimum. It has been estimated that the worst case scenario gives a maximum of 5 fragments. Figure 5 shows how the fragmentation level is related to n.

Table 5: Worst case scenarios for fragmentation.

n	**Fragments**	**Method Rounds**
1024 bits	0	4
1536 bits	0	4
2048 bits	0	4
4096 bits	≤ 2	≤ 6
6144 bits	≤ 2	≤ 6
8192 bits	≤ 5	≤ 9

3.3.6 Identity Protection

The SRP-6 protocol allows client's identity (I) to be sent in clear over the communication channel, without compromising protocol security. Nevertheless, EAP-SRP-256 adds an identity-hiding mechanism in order to meet RFC 4017 requirements and increase the effort for the attackers to obtain userIDs. To do so the EAP method uses pseudorandom values (id_value) that are derived from the exchanged encrypted data and work as pseudonyms. They are different for each session.

3.3.7 Pseudorandom Numbers Generator

The generation of pseudorandom numbers is required during the entire authentication process. EAP-SRP-256 supports a PRNG algorithm based on the ANSI-X9.31 (Keller, 2005) and FIPS-140-2 specifications. The two major advantages are the adoption of a standard technique and the reuse of the available cryptographic primitive.

The algorithm uses AES with ${}^*K$, a secret key reserved for number generation, DT, a date-time vector, and an arbitrary initialization seed V that must be kept secret. An iterative sequence could create pseudorandom values R computing:

$$I = \mathrm{AES}_{{}^*K}(DT) \tag{4}$$

$$R = \mathrm{AES}_{{}^*K}(I \oplus V) \tag{5}$$

$$V = \mathrm{AES}_{{}^*K}(R \oplus I) \tag{6}$$

The support to this PRNG is optional but recommended by method specifications. For other solutions, implementators should consider RFC 4086 (Eastlake et al., 2005) guidelines.

3.4 Security

The expected security level of EAP-SRP-265 has been evaluated by considering the building blocks of its architecture and known attacks on similar protocols. Security considerations have covered:

- the SRP-6 protocol;
- key generation and management;
- mathematical properties of the parameters;
- Man-in-The-Middle attacks;
- Replay attacks;
- Dictionary attacks;
- Packet modification attacks;
- compliance with RFC 4017 security requirements.

According to the provided analysis, the proposed EAP method owns formal and cryptographic properties that enable it to work correctly. If compared to other popular solutions (Figure 3), it is believed to offer an adequate security level for wireless networks giving some advantage such as PKI independence and a provable robustness for key derivation.

3.5 Prototype

A working C/C++ implementation[2] of EAP-SRP-256 has been developed for Wi-Fi networks in order to study protocol applicability to real environments. This prototype provides the integration with *freeRADIUS*, a popular RADIUS authentication Server, and *Xsupplicant*, a client-side IEEE-802.11i implementation.

FreeRADIUS rappresents today a widespread solution in many environments. The development on server side is conceived just like an EAP module, while the client side has been directly integrated in the application.

The programming library *Libgcrypt* has been used to perform the cryptographic operations such as AES-256, SHA-256, HMAC and all the modular calculus. It is in part derived from the GNU Multi-Precision Library (GMP) and used primarily by the GNU Privacy Guard (GPG) software. This library uses many assembler implementations of very low level functions to gain much better performance than with the standard C implementation.

[2]Publicly available under GPL license at `http://sourceforge.net/projects/eap-srp-256/`

	EAP-TLS	PEAP	EAP-SRP-256
Authentication Type	PKI-Based	Tunneled	Password-Based
Basic operations	TLS Handshake with certificate exchange.	TLS Handshake and Encrypted Tunnel	SRP-6 protocol and exchange of encrypted parameters
Inner Protocols	TLS	TLS + EAP-MSCHAPv2 TLS + EAP-GTC	SRP-6
Primitives	Negotiable	Negotiable	SHA-256 AES-256
Rounds	4	6	4
Key derivation mechanism	PRF	PRF + Cryptographic Binding	Modified Counter Mode
Key Strength	Variable	Variable	256 bit
Message fragmentation support	Yes	Yes	Yes
Certificates	X.903 v3 (both sides)	X.903 v3 (only Server-side)	Not required
Identity-Protection	No	Yes	Yes
Integrity-Protection	Yes	Yes	Yes
Protected Ciphersuite Negotiation	Not required	Yes	Not required
Fast-Reconnect	No	Yes	Under evaluation

Figure 3: Comparison with other methods.

4 CONCLUSIONS AND FUTURE WORKS

In this paper we presented EAP-SRP-256, a new authentication method proposal designed for wireless networks that support the Extensible Authentication Protocol (EAP). The proposed method mainly relies on the SRP-6 protocol and it provides mutual authentication using a strong password-based scheme. The given definition wants primarily to be compliant with IETF security and key management requirements for the EAP methods.

At present, work is being done to develop a formal analyis for the proposed EAP method. The main purpose is to apply a model theoretic definition that can demonstrate protocol correctness also from the mathematical standpoint. In addition, the available Wi-Fi implementation is being deployed in real world environments in order to analyze protocol behavior and performance related to the current specifications.

Future work involves investigating the possibility of further protocol enhancements. Features like the *Fast-reconnect* option are already under evaluation. This would permit clients to re-authenticate using an alternate protocol exchange with a reduced round

number and a lower computation overhead. Another research area is the extendibility to a general protocol model that provides a negotiation mechanism for crypto primitives and hash functions.

Finally there is a need do develop an implementation for WiMax networks in order to demonstrate the applicability of EAP-SRP-256 to different communication systems.

REFERENCES

Aboba, B., Blunk, L., Vollbrecht, J., and Carlson, J. (2004). Extensible authentication protocol (EAP). RFC 3748. (Obsoletes RFC 2284).

Aboba, B. and Simon, D. (1999). PPP EAP TLS authentication protocol. RFC 2716.

Bellare, M., Pointcheval, D., and Rogaway, P. (2000). Authenticated key exchange secure against dictionary attacks. *Lecture Notes in Computer Science*, 1807:139.

Bellare, M. and Rogaway, P. (2000). The AuthA protocol for password-based authenticated key exchange. Technical report. Contribution to the IEEE P1363 study group for Future PKC Standards.

Bernard Aboba, e. a. (2006). Extensible authentication protocol (EAP) key management framework. IETF Internet draft (Work in Progress).

Bersani, F. and Tschofenig, H. (2007). The EAP-PSK protocol: A pre-shared key extensible authentication protocol (EAP) method. RFC 4764.

Daemen, J. and Rijmen, V. (2002). *The Design of Rijndael*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. ISBN 3540425802.

Diffie, W. and Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654.

Dobbertin, H., Knudsen, L. R., and Robshaw, M. J. B. (2004). The cryptanalysis of the AES - a brief survey. In *AES Conference*, pages 1–10.

Dolev, D. and Yao, A. C. (1981). On the security of public key protocols. Technical report, Stanford, CA, USA.

Dworkin, M. (2001). Recommendation for block cipher modes of operation - methods and techniques. NIST Special Publication 800-38A, National Institute of Standards and Technology.

Eastlake, D., Schiller, J. I., and Crocker, S. (2005). Randomness requirements for security. RFC 4086.

Ferguson, N. and Schneier, B. (2003). *Practical Cryptography*. Wiley Publishing Inc. ISBN 0-471-22894-X.

Funk, P. (2005). EAP tunneled TLS authentication protocol version 0 (EAP-TTLSv0). IETF Internet draft (Work in Progress).

Gilbert, H. (2003). The security of one-block-to-many modes of operation. *Springer-Verlag LNCS*, FSE 03(2287):376–395. ISBN 3-540-20449-0.

Heintze, N. and Tygar, J. D. (1996). A model for secure protocols and their compositions. *Software Engineering*, 22(1):16–30.

Hoffman, P. and Schneier, B. (2005). Attacks on cryptographic hashes in internet protocols. RFC 4270.

Keller, S. S. (2005). NIST-Recommended random number generator based on ANSI X9.31 Appendix A.2.4 using the 3-key triple DES and AES algorithms. NIST Information Technology Laboratory - Computer Security Division, National Institute of Standards and Technology.

Koblensky, M. (2006). Implementazione del protocollo di autenticazione EAP-SRP-256. *Master Thesis at the Dipartimento di Sistemi e Informatica, Universita' di Firenze, Italy*.

Krawczyk, H., Bellare, M., and Canetti, R. (1997). HMAC: Keyed-hashing for message authentication. RFC 2104.

Luby, M. and Rackoff, C. (1988). How to construct pseudorandom permutations from random functions. *SIAM J. Computing*, Vol. 17 No. 2.

Manganaro, A. (2005). Studio di un metodo di autenticazione per le reti wireless basato sul protocollo SRP-6. *Master Thesis at the Dipartimento di Sistemi e Informatica, Universita' di Firenze, Italy*.

Millen, J. and Shmatikov, V. (2003). Symbolic protocol analysis with products and diffie-hellman exponentiation. In *Proceedings of the 16th IEEE Computer Security Foundations Workshop.*, Asilomar, USA.

Palekar, A., Simon, D., Salowey, J., Zhou, H., Zorn, G., and Josefsson, S. (2004). Protected EAP protocol (PEAP) version 2. IETF Internet draft (Work in Progress).

Skoudis, E. (2002). *Counter Hack - A step-by-step Guide to Computer Attacks and Effective Defenses*. Prentice Hall PTR. ISBN 0-13-033273-9.

Stanley, e. a. (2005). EAP method requirements for WLAN. RFC 4017.

Taylor, D., Wu, T., Mavrogiannopoulos, N., and Perrin, T. (2006). Using SRP for TLS authentication. IETF Internet draft (Work in Progress).

Wagner, D. and Schneier, B. (1996). Analysis of the SSL 3.0 protocol. In *Proceedings of the Second USENIX Workshop on Electronic Commerce*, Oakland, California.

Wu, T. (1997). The secure remote password protocol. In *Proceedings of the 1998 Internet Society Network and Distributed System Security Symposium*, pages 97–111, San Diego, CA.

Wu, T. (October 2002). SRP-6: Improvements and refinements to the secure remore password protocol. *Submission to the IEEE P1363 Working Group*.

Zhao, Z., Dong, Z., and Wang, Y. (2006). Security analysis of a password-based authentication protocol proposed to IEEE 1363. *Theor. Comput. Sci.*, 352(1):280–287.