Key Exchange in IPsec revisited: Formal Analysis of IKEv1 and IKEv2

Cas Cremers

ETH Zurich, Switzerland
cas.cremers@inf.ethz.ch

Abstract. The IPsec standard aims to provide application-transparent end-to-end security for the Internet Protocol. The security properties of IPsec critically depend on the underlying key exchange protocols, known as IKE (Internet Key Exchange).
We provide the most extensive formal analysis so far of the current IKE versions, IKEv1 and IKEv2. We combine recently introduced formal analysis methods for security protocols with massive parallelization, allowing the scope of our analysis to go far beyond previous formal analysis. While we do not find any significant weaknesses on the secrecy of the session keys established by IKE, we find several previously unreported weaknesses on the authentication properties of IKE.

Keywords: Security protocols, IPsec, IKE, IKEv1, IKEv2, Formal analysis, protocol interaction, multi-protocol attacks

1 Introduction

IPsec [19] is an IETF protocol suite that provides Internet Protocol (IP) security. In particular, IPsec provides confidentiality, data integrity, access control, and data source authentication [17]. In contrast to, e. g., SSL/TLS [12], IPsec provides end-to-end security in an application-transparent way, i. e., without having to change each application separately. IPsec was developed in conjunction with IPv6, and any compliant implementation of IPv6 must support IPsec. For IPv4, IPsec is an optional extension.

The main goal of IPsec is to provide for each packet a means to guarantee origin authentication, integrity, and confidentiality. Additionally, IPsec may provide identity protection for the communicating parties. IPsec implements this by setting up for each communication a so-called *Security Association*, which essentially boils down to establishing a session key, which is used to provide the three main security properties for the subsequently transmitted messages. Perhaps surprisingly, establishing such a session key in IPsec is an involved process with a large amount of options to choose from, such as various sub-protocols, cryptographic methods, and optional fields. The protocol suite responsible for this key establishment phase is known as IKE (Internet Key Exchange). According to its specification, IKE performs "mutual authentication between two parties and establishes an IKE security association" [17].

Currently, there are two versions of IKE. The original design, IKEv1 [15], was criticised for its complexity and large amount of options. Its successor, IKEv2 [16, 17], is significantly simpler and seems to provide at least the same level of security, but still offers a large amount of options.

When IKEv1 was proposed, it was analyzed by several groups, e. g., [14, 25, 29, 30, 32]. Given that recent years have seen many new developments in methods to analyze security protocols, as well as the establishment of many strong security notions for key exchange protocols, one might expect that IKE is a prime candidate for an updated analysis. Surprisingly, only limited analysis was performed, e. g., [26] for some sub-protocols of IKEv2.

Contributions. We provide the most comprehensive formal analysis of IKE so far. This includes providing the first complete analysis of unintended interactions between pairs of protocols in IKE, and analyzing IKE with respect to formalizations of advanced cryptographic security notions [5, 22]. We find a large number of previously unreported logical weaknesses. In general, these weaknesses do not lead to violations of the main security goals, but show that strong authentication properties are not always achieved. Our analysis thus provides a more precise view on the security properties provided by the IKE protocols.

We provide full details of our tests and protocol models online [10].

Overview. We give background on the IKE protocols and describe previous analyses of IKE in Section 2. We describe the setup and details of our analysis in Section 3. In Section 4 we provide an overview of the results of our analysis and describe both rediscovered and newly found weaknesses. Possible fixes and remaining issues are discussed in Section 5. Finally, we conclude in Section 6.

In the Appendix we describe the used adversary models and give the full analysis results for multi-protocol attacks.

Acknowledgements. The author is grateful to Adrian Kyburz, whose IKE models and initial analysis [21] provided the starting point for this work.

2 Background on IKE

In this section we provide background on the IKEv1 and IKEv2 protocols and their intended security properties. Additionally we give an overview of related work, in particular previous analyses of these protocol standards.

A *Security Association* (SA) is the establishment of a secure transmission session between two network entities. This amounts to establishing a set of shared attributes, which typically include the chosen cryptographic algorithms and one or more keys for encrypting/decrypting traffic. The main purpose of IKE within IPsec is to establish a security association, more specifically the *IPsec SA*, between two authenticated IPsec peers. The IPsec SA includes traffic keys that can be used for a secure IPsec tunnel.

Notation. We denote the (plaintext) headers in each IKE message by HDR_i for some i. We write $\{m\}_{sk(X)}$ to denote the digital signature of the agent X of the message m, and $\{m\}_{pk(X)}$ for the public key encryption of m with X's public key. For a key K that is not of the form $sk(X)$ or $pk(X)$, $\{m\}_K$ denotes the symmetric encryption of m with key K. Fresh random values, usually referred to as Nonces, are denoted as N_X when created by the agent X. ID_X denotes (a representation of) the identity of agent X. The short-term Diffie-Hellman private key of agent A, also known as the ephemeral private key, is written as x_A and the corresponding ephemeral public key is written as g^{x_A}. We write $prf_K(m)$ to denote the application of the pseudorandom function prf to the message m, optionally keyed with K. CKY_A and CKY_B respectively denote the initiator's and responder's cookie, which are also included in the headers. Following the standard, we denote the proposals for negotiating security parameters by SA in the protocol descriptions.

2.1 IKE version 1 (IKEv1)

The design of IKEv1 [15] is based on the Oakley protocol [27] and ISAKMP [24]. The protocol is essentially an authenticated key exchange protocol with additional payloads that supports multiple cryptographic algorithms and which is split into two distinct phases. In phase 1 an ISAKMP SA is established that is used in phase 2 to set up an IPsec SA.

In phase 1, the peers are authenticated and a shared secret key is established. The shared secret key is derived from the exchanged Diffie-Hellman tokens. Phase 1 can be performed in one of the following two modes: *Main Mode* (MM) and *Aggressive Mode* (AM). Main Mode provides *identity protection* (i. e., an adversary cannot determine from the messages who the participating agents are) and only exchanges security parameters with authenticated peers. The Main Mode protocol consists of six messages. In contrast, Aggressive Mode offers no identity protection and may exchange some security parameters with peers before they are authenticated, but requires only three messages.

For both MM and AM, IKEv1 supports three types of underlying key infrastructures, based on respectively symmetric cryptography, digital signatures, and public key cryptography. For public key cryptography there are two different types of protocol, one of which is supposed to replace the other but has been retained for compatibility purposes. The combination of the two modes with the four key types yields eight phase 1 protocols to establish a shared key.

Example 1 (IKEv1 Aggressive Mode with digital signatures). The IKEv1 AM protocol with digital signatures proceeds as follows:

$$
\begin{aligned}
&1.\ A \to B : \mathrm{HDR}_1,\ SA, g^{x_A}, N_A, ID_A \\
&2.\ B \to A : \mathrm{HDR}_2,\ SA, g^{x_B}, N_B, ID_B, \\
&\qquad\qquad \{prf_K(g^{x_B}, g^{x_A}, CKY_B, CKY_A, ID_B)\}_{sk(B)} \\
&3.\ A \to B : \mathrm{HDR}_3,\ \{prf_K(g^{x_A}, g^{x_B}, CKY_A, CKY_B, ID_A)\}_{sk(A)}
\end{aligned}
$$

where K is a shorthand for $prf_{(N_A,N_B)}(g^{x_A x_B})$.

After the completion of phase 1, the resulting shared secret is used to derive several shared secret keys. In what follows, these are denoted by SKEYID_x where x identifies the specific key.

The shared secret keys established in phase 1 are used in phase 2 to establish or update the session keys for the IPsec tunnel. This second phase is called *Quick Mode* (QM), and consists of three protocols. The first protocol provides perfect forward secrecy (PFS),i. e., revealing the long-term keys does not compromise the security of past sessions, but no identity protection. The second provides no perfect forward secrecy but is more efficient than the first, and the third provides identity protection. In IKEv1, AM and MM are always directly followed by QM.

Example 2 (IKEv1 Quick Mode without perfect forward secrecy). Let M-ID denote a message identifier and let KAB and KBA be the appropriate symmetric traffic encryption keys derived from the secret established in phase 1.

$$
\begin{array}{ll}
1. & A \to B : \text{HDR}_1, \{\ prf_{\text{SKEYID}_a}(\text{M-ID}, SA, N'_A), SA, N'_A\ \}_{KAB} \\
2. & B \to A : \text{HDR}_2, \{\ prf_{\text{SKEYID}_a}(\text{M-ID}, N'B, SA, N'_A), SA, N'_B\ \}_{KBA} \\
3. & A \to B : \text{HDR}_3, \quad prf_{\text{SKEYID}_a}(0, \text{M-ID}, N'_A, N'_B)
\end{array}
$$

The resulting new keying material is defined as $prf_{\text{SKEYID}_\text{d}}(\text{protocol}, SPI, N'_A, N'_B)$, where protocol and SPI are taken from the ISAKMP SA proposals.

2.2 IKE version 2 (IKEv2)

IKEv2 [16, 17] was designed to add new features, address some weaknesses in IKEv1 and, at the same time, provide a cleaner design. In IKEv2 there are only three sub-protocols in phase 1. These are based on digital signatures, MAC's, and EAP (Extensible Authentication Protocol). Similar to IKEv1, Diffie-Hellman exponents and nonces are exchanged and used to compute several shared secret keys. For example, the initiator i will use SK_{ei} (for encryption), SK_{ai} (for authentication), and SK_{pi} (for the authentication payload). Similarly, the responder r will use SK_{er}, SK_{ar}, and SK_{pr}.

The phase 1 protocols establish an IKE SA (similar to IKEv1's ISAKMP SA) and a first child SA (similar to IKEv1's IPsec SA). Hence, contrary to IKEv1, an SA that can be used for IPsec (the child SA) is directly available after the first phase. The specification of the first phase protocols contains optional fields, and allows for initiators and responders to use different authentication mechanisms.

Example 3 (IKEv2 phase 1 with digital signatures). Let SAi1 and SAr2 denote the supported cryptographic algorithms and let SAi2 and SAr2 denote IPsec SA proposals. In this example, we write $\{m\}_{SK_x}$ to denote that m is integrity protected and encrypted using SK_{ax} and SK_{ex}. TSi and TSr denote traffic selectors and are not directly relevant for our analysis. Similarly, $SPIi$ and $SPIr$ denote Security Parameter Indexes. Furthermore, we define

$$
\begin{aligned}
AUTHi &= \{\text{SAi1}, g^{x_A}, N_A, N_B, prf_{SK_{pi}}(ID_A)\}_{sk(A)} \\
AUTHr &= \{\text{SAr1}, g^{x_B}, N_B, N_A, prf_{SK_{pr}}(ID_B)\}_{sk(B)}
\end{aligned}
$$

Then, the protocol proceeds as follows:

1. $A \to B : \mathrm{HDR}_1, \mathrm{SAi1}, g^{x_A}, N_A$
2. $B \to A : \mathrm{HDR}_2, \mathrm{SAr1}, g^{x_B}, N_B$
3. $A \to B : \mathrm{HDR}_3, \{ID_A, ID_B, AUTHi, \mathrm{SAi2}, TSi, TSr\}_{SK}$
4. $B \to A : \mathrm{HDR}_4, \{ID_B, AUTHr, \mathrm{SAr2}, TSi, TSr\}_{SK}$

The second phase in IKEv2 is known as *Child Mode*. The purpose of Child Mode is to re-key previous (IKE or child) SA's or to establish additional child SA's. For Child Mode, there are only two protocols, one of which provides Perfect Forward Secrecy. These are similar to their IKEv1 phase 2 counterparts, but consist only of two messages. For further details we refer to [17].

2.3 Intended security properties

The IKE standards mention the following intended properties of the protocols:

1. To obtain authenticated keying material for use with ISAKMP and for other security associations [15, p. 1].
2. To achieve the security goals provided by Oakley [27]: perfect forward secrecy for keys, identity protection, and authentication [15, p. 2].
3. To perform mutual authentication [17, p. 5].

The standards do not contain more detailed descriptions of these properties.

2.4 Previous analyses of IKE

IKEv1 has been analyzed before by several authors. In 1999, Meadows performed a large case-study of IKEv1 using the NRL protocol analyzer [25], a formal protocol analysis tool. Her analysis uncovered many subtle properties and weaknesses at the logical level. In 2000, Perlman and Kaufman performed a manual analysis of IKEv1 [29, 30]. Their analysis covered not only the logical level, but also many different other aspects of the protocol, leading to a significant critique of the standard. Many of their suggestions were included in the design of the IKEv2 standard. Zhou analyzed further issues of the IKEv1 standard [32] and suggests amendments for the weaknesses he finds. In 2001 and 2002, Canetti and Krawczyk analyzed selected sub-protocols of IKE in the cryptographic setting [5, 6], e. g., showing that a variant of IKE Main Mode with signatures satisfies a form of cryptographic key-exchange security.

In contrast, there has only been very limited analysis of IKEv2. In 2003, three sub-protocols of IKEv2 were formally analyzed in the context of the AVISPA project [1, 26]. They found that an IKEv2 sub-protocol suffers from a weakness that was already pointed out on a related IKEv1 protocol by Meadows.

3 Formal analysis of IKEv1 and IKEv2

In this section we describe the setup of our analysis. The results will be described in Section 4. Similar to the approach taken by Meadows for her analysis of IKEv1 [25], we follow the line of work by Dolev and Yao [13]. Hence we focus on the detection of logical vulnerabilities of protocols, and assume that cryptography is perfect in the sense that, for example, the adversary learns nothing from an encrypted message unless he knows the decryption key. This can be seen as a separation of concerns: we rely on the stated properties of the used cryptographic algorithms. A second assumption is that the adversary has full control over the network, and can intercept or modify all messages, or inject his own.

Our analysis covers significantly more security aspects than the earlier formal analyses of IKE. Along the lines of the security notions for key exchange first explored by Bellare and Rogaway [3], we also consider various advanced security properties, such as perfect forward secrecy, key compromise impersonation, and known-key attacks. Here we follow the formalizations by Basin and Cremers [2]. Additionally, we consider interactions between sub-protocols, which are an instance of multi-protocol attacks [11, 18].

Formal analysis tool and extensions. In our analysis we used Scyther [8], a formal protocol analysis tool. In particular, we exploited two of its features: analyzing protocol properties with respect to various adversary models [2], and support for multi-protocol analysis [11].

For the analysis of IKE, we extended the Scyther tool with two features. First, we provided support for checking *agreement* as defined by Lowe [23]. Scyther's built-in notions of authentication, (weak) aliveness and synchronisation[1], were respectively too weak and too strong in the context of IKE. Second, because the scope of our analysis is currently not feasible on standard computing hardware, we developed infrastructure for performing large-scale *parallel* analysis for multi-processor computing clusters. We will return to this issue in Section 3.

Protocol models. A critical step in the formal analysis consists of providing abstract models of the protocols under investigation, such that they can be analyzed within the formal framework underlying the tool [9]. The abstract protocol models should include all security-relevant information. It is critical to find a balance here between precision (i. e., complexity) of the models and feasibility of the analysis.

For our models, we abstracted the security association proposals into arbitrary choices (but checking that the recipient and sender agree on the contents of the proposal). Similarly, we abstracted from the public key certificates and assume that they have been pre-distributed in a secure manner, which is justifiable as our models do not consider concepts such as key revocation.

Within the formal framework, the commutativity property of the Diffie-Hellman exchange ($(g^a)^b = (g^b)^a$) currently cannot be modeled precisely, and we

[1] This can be considered similar to the notion of "matching histories", and requires messages to have been sent exactly as received, as well as in the correct order.

name	mode	encrypt last message	separate encryptions
ikev1-sig-m	MM		
ikev1-sig-a1	AM	N	
ikev1-sig-a2	AM	Y	
ikev1-pk-m	MM	N	Y
ikev1-pk-m2	MM	N	N
ikev1-pk-a1	AM	N	Y
ikev1-pk-a12	AM	Y	Y
ikev1-pk-a2	AM	N	N
ikev1-pk-a22	AM	Y	N
ikev1-pk2-m	MM	N	Y
ikev1-pk2-m2	MM	N	N
ikev1-pk2-a	AM	N	Y
ikev1-pk2-a2	AM	N	N
ikev1-psk-m	MM		
ikev1-psk-a	AM		
ikev1-quick	QM		
ikev1-quick-noid	QM		
ikev1-quick-nopfs	QM		
ikev1-sig-a-perlman1	AM	N	
ikev1-sig-a-perlman2	AM	Y	
ikev1-sig-m-perlman	MM		
ikev1-psk-m-perlman	MM		

Table 1. Overview of IKEv1 protocol models. The bottom four protocols are not part of the standard, but are improvements suggested in [30]

name	mode	optional identity
ikev2-sig	SIG	Y
ikev2-sig2	SIG	N
ikev2-mac	MAC	Y
ikev2-mac2	MAC	N
ikev2-eap	EAP	Y
ikev2-eap2	EAP	N
ikev2-sigtomac	SM	Y
ikev2-sigtomac2	SM	N
ikev2-mactosig	SM	Y
ikev2-mactosig2	SM	N
ikev2-child	C	
ikev2-child-nopfs	C	

Table 2. Overview of IKEv2 protocol models

therefore underapproximate this property by giving the adversary the capability of rewriting such exponentiations at fixed subterm positions, which are derived from the protocol specification. Full details can be found in the protocol models. Our underapproximation ensures soundness of the attacks that we find.

In Table 1 we list the models of the sub-protocols that we consider for IKEv1. The first column shows the names which are meant to be self-explanatory as far as possible, including the underlying cryptographic mode in the name (hence pk and pk2 for the two public-key variants, and sig and psk for signatures and pre-shared keys, respectively). The second column shows the mode. For phase 1 we have "MM" for Main Mode and "AM" for Aggressive Mode. For phase 2 we have "QM" for Quick Mode. The third column, "encrypt last message", marks protocol variants in which the last message is encrypted by the session key. The fourth column, "separate encryptions", refers to an ambiguity in the specification: in one message, it is required that a nonce and an ID are encrypted. This can

be interpreted in two ways: first encrypt each separately, and then concatenate the result, or alternatively, first concatenate the nonce and ID and then encrypt the result. For our analysis, we have modeled both interpretations. For Quick Mode, there are three variants. The first two are Diffie-Hellman based exchanges with and without the optional identity fields. The third variant uses plain nonces instead of Diffie-Hellman and can be used when perfect forward secrecy (PFS) is not required. Besides the protocol models described in the IKEv1 and IKEv2 standards, we also modeled four variants suggested by Perlman and Kaufman.

Similarly, Table 2 lists the sub-protocols modeled for IKEv2. The main modes here are "SIG", "MAC", and "EAP", denoting digital signatures, MACs, and Extensible Authentication Protocol (EAP), respectively. Additionally, "SM" denotes the variants in which one agent uses a different mode than the peer within the same sub-protocol. Finally, "C" denotes Child Mode, which is IKEv2's variant of the re-keying phase. The third column marks the variants in which optional identity fields are omitted in the specification.

Our full protocol models are publicly available at [10].

Security properties. As stated before, the IKE standards do not provide detailed descriptions of the intended properties. One of the contributions of our work is therefore to establish more precisely which properties are guaranteed by the protocols. In our analysis, we consider the following *basic security properties*.

Aliveness If an agent a executes a role of the protocol, thinking he ran it with b, then b has indeed performed an action (and is therefore "alive"). This is a very weak authentication property.

Weak agreement Aliveness, where additionally b assumes that he is communicating with a, and hence they both "agree" on the agents involved in the protocol.

Agreement (on a list of terms S) This implies weak agreement and additionally, b is indeed performing the role that a assumes. Finally, a and b agree on the values in S, e. g., they agree on the computed session key.

Secrecy (of a term t) The term t, e. g., a computed session key, will not become known to the adversary.

More formally, for the first three properties, we follow the definitions given in [23], and for the final property, we follow [2]. The supposed security properties are specified in the protocol description by means of *claim events*, such that they can be analyzed by the Scyther tool.

Following [2], we combine basic security properties with *adversary models*. We consider a total of 10 adversary models, which include the standard Dolev-Yao model, as well as abstract versions of the adversary capabilities considered in the CK model [5] and the eCK model [22], which are commonly used security models for proving the security of authenticated key exchange protocols. We informally describe these models in Appendix A.

Combining the above basic security properties with the appropriate adversary model, we analyze the protocols with respect to the following properties.

(Perfect) Forward Secrecy We consider secrecy of a term (typically a session key) occurring in a session s whilst allowing the adversary to compromise the long-term keys of all agents after s has been completed. Thus, the compromise of the long-term keys should not lead to the compromise of session keys (or any session-specific secret) from earlier sessions.

Weak Perfect Forward Secrecy Similar to Perfect Forward Secrecy above, but slightly weaker: if he did not actively interfere in the session s, then he may compromise the long-term keys of agents. This notion was introduced in [20] as a variant of Perfect Forward Secrecy suitable for a class of implicitly authenticated key exchange protocols, such as HMQV.

(resilience to) Key Compromise Impersonation If the adversary is able to compromise the long-term keys of an agent a, he should not be able to impersonate as an arbitrary agent to a. This can be modeled by analyzing authentication properties in the presence of an adversary capable of compromising the actor's long-term keys.

(resilience to) known session key attacks If the adversary learns a session key, this should not allow him to attack other sessions. Here we follow Bellare and Rogaway [3] and analyze secrecy of a particular session key in the presence of an adversary that can compromise all session keys of *non-matching sessions*. We will return to the concept of matching sessions when discussing the results.

Using the same mechanism, i. e., combining basic security properties with adversary models, allows us to analyze protocols with respect to abstract versions of cryptographic security notions, such as the previously mentioned CK and eCK models. Full details and formal specifications of the protocol execution model and the various adversary capabilities can be found in [2].

Multi-protocol attacks. We also considered possible protocol interactions in the context of a standard Dolev-Yao adversary (i. e., with static corruption). When (sub-)protocols that are executed in parallel share the same long-term keys, the interaction between the protocols can enable *multi-protocol attacks* [11, 18]. As already pointed out in [25], given the many sub-protocols in IKE, unforeseen protocol interactions cannot be excluded. However, analysing all possible interactions is significantly harder than their individual analysis, and therefore not all possible interactions were considered by Meadows in [25].

In our analysis we consider all possible interactions between pairs of subprotocols in IKEv1 and IKEv2. In theory, attacks could be possible that require interactions among three or more protocols [11] but this is currently infeasible to analyze in our setup.

Analysis hardware. Our analysis would not have been feasible without using a significant amount of hardware. The enabling factor for our analysis was the high-performance cluster of ETH Zurich, called *Brutus* [31]. Brutus is a heterogeneous system with several types of compute nodes. Currently, Brutus has a total of 9912 processor cores in 1108 compute nodes. The majority of the compute nodes are four quad-core AMD Opteron 8380 CPUs with 32 GB of RAM. In our tests we used a maximum of 128 processor cores in parallel, due to default usage limits.

Our analysis mostly consisted of a large number of relatively small tasks, such as analyzing a single protocol property with respect to a particular adversary model, and hence could be easily parallelized. The set of tests for analyzing all protocols with respect to all adversary models, excluding the multi-protocol analysis, took about a day of computation on the cluster. The multi-protocol analysis for all two-protocol combinations with respect to a single adversary model took just over two days. To put these numbers into perspective, on a single desktop machine these computations would have taken at least two orders of magnitude more time, thus requiring about a year of computation time. Of course, in practice, the situation is worse: in some cases the test results revealed modeling errors, whose fixing required partial recomputation of the results. Thus, we used the Brutus cluster therefore much longer than three days, and analysis using a single machine would not have been feasible.

4 Results

Our analysis automatically rediscovers known weaknesses but also discovers many new weaknesses, as shown in Tables 3 and 4. We briefly discuss the rediscovered weaknesses, before explaining in detail the newly discovered weaknesses.

Protocol	Roles	Violated properties	Classification
ikev1-pk-m2	I	Agreement, Weakagree (†)	N1
ikev1-pk2-m2	I	Agreement, Weakagree (†)	N1
ikev1-psk-m	I	Agreement, Weakagree (†)	K1
ikev1-psk-m-perlman	I	Agreement, Weakagree (†)	K1
ikev1-quick-noid	I,R	Aliveness, Agreement, Weakagree	K2
ikev1-quick-nopfs	I,R	Aliveness, Agreement, Weakagree	K2
ikev1-sig-a-perlman1	I	Agreement, Weakagree	N2
ikev1-sig-a-perlman2	I	Agreement, Weakagree	N2
ikev1-sig-a1	I	Agreement, Weakagree	N2
ikev1-sig-a2	I	Agreement, Weakagree	N2
ikev1-sig-m	I	Agreement, Weakagree (†)	K1
ikev1-sig-m	R	Agreement, Weakagree	N3
ikev1-sig-m-perlman	I	Agreement, Weakagree	K1

Table 3. Overview of attacks on properties of IKEv1 sub-protocols in the presence of a standard Dolev-Yao adversary. A dagger (†) denotes that the attacks require self-communication.

4.1 Automatically rediscovered weaknesses

(K1) Reflection attack on IKEv1 Main Mode with digital signatures or pre-shared keys. In [14], Ferguson and Schneier report a reflection attack

Protocol	Roles	Violated properties	Classification
ikev2-child	I,R	Aliveness, Agreement, Weakagree	N4
ikev2-child-nopfs	I,R	Aliveness, Agreement, Weakagree	N4
ikev2-sig2	R	Agreement, Weakagree	K3
ikev2-sigtomac2	R	Agreement, Weakagree	K3

Table 4. Overview of attacks on properties of IKEv2 sub-protocols with respect to a Dolev-Yao style adversary (INT).

on IKEv1 Main Mode when used with digital signatures or pre-shared keys. The attack is a simple reflection that requires that an initiator may accept her own identity as the peer. We will refer to the case in which an agent accepts her own identity as the peer as the *self-communication* scenario. In this case, the authenticators in the protocol become equal, and the adversary simply sends all messages coming from the initiator back to her, unchanged. In this case weak agreement fails, because there is no agent performing the responder role. Secrecy of the key is still guaranteed. Such a self-communication scenario may be relevant under some circumstances, as argued, e. g., in [4]. However, it is clear that self-communication is not prohibited by the specification of IKEv1 and should therefore be considered a legitimate use. When self-communication is not possible, e. g., because identity-inequality checks are implemented for both roles, the attack pointed out by Ferguson and Schneier is no longer possible.

(K2) Reflection attack on IKEv1 Quick Mode. Meadows reported a reflection attack on IKEv1 Quick Mode [25]. Because of the symmetry of the messages, and the possibility of leaving out some optional identities, the recipients of messages cannot determine directly whether they sent these messages themselves (possibly in other sessions). This leads to a straightforward reflection attack, and even allowing the reflection of messages to the responder role. Meadows observed that this may be prevented by other details of the implementation, but it is clear that there is no mechanism at the logical level to prevent reflections, and hence this may be an issue in some implementations.

(K3) Weak authentication for IKEv2 with digital signatures. The IKEv2 digital signature mode allows identities to be omitted. As a result, an agent may successfully complete the responder role while her peer believes that he is in a partial run, talking to a different agent [26]. This constitutes a violation of agreement on the agent identities. Moedersheim et al. report this as a violation of "penultimate authentication" (We will return to this property in Section 5). We observe that this violates strong authentication for the responder. In practice this means that an IKEv2 responder Bob may accept an IPsec SA as valid for Alice, even though only weak authentication is provided: it may not be the case that Alice intended to talk to Bob. However, subsequent messages received over

the IPsec tunnel are guaranteed to provide strong authentication. We will see an example of a similar weakness below (N2).

4.2 Previously unreported weaknesses

(N1) Reflection attack on IKEv1 Main Mode with public key encryption. Our analysis reveals that the reflection attack (K1) reported by Ferguson and Schneier, is also possible for the public key variants. Similar to their attacks, the self-communication scenario is required for the attacks to work.

(N2) Authentication failure on IKEv1 Aggressive Mode with digital signatures. For IKEv1 Aggressive Mode with digital signatures, as described in Example 1, we find that (weak) agreement is not guaranteed for the initiator. The following scenario can occur:

1. A generates x_A and N_A and sends (for B): $\mathrm{HDR}_1, SA, g^{x_A}, N_A, ID_A$.
2. The adversary intercepts this message and changes ID_A to ID_C for an arbitrary agent C. The adversary sends the result to B: $\mathrm{HDR}_1, SA, g^{x_A}, N_A, ID_C$.
3. B accepts the message, and assumes that g^{x_A} and N_A were generated by C.
4. B generates x_B and N_B and computes the key $K = prf_{(N_A,N_B)}((g^{x_A})^{x_B})$.
5. B sends the following message for C:
   $\mathrm{HDR}_2, SA, g^{x_B}, N_B, ID_B, \{prf_K(g^{x_B}, g^{x_A}, CKY_B, CKY_A, ID_B)\}_{sk(B)}$.
6. The adversary intercepts this message and sends it to A, who accepts it, computing the same key K.
7. A sends (for B): $\mathrm{HDR}_3, \{prf_K(g^{x_A}, g^{x_B}, CKY_A, CKY_B, ID_A)\}_{sk(A)}$.
   Note that this message is not what B expects, and will be rejected by B.
8. A successfully completes her part of the protocol.

Consequently, agreement on the participants is not provided by this protocol: after A finishes her initiator role with B, B was indeed running the responder role, but B was under the assumption he was running the protocol with C. When A finishes her role, B may still be waiting for the final message, or may have aborted. Clearly, B will not accept the final message because he is expecting the message to be signed by C. However, A cannot detect whether B has accepted the final message unless additional messages are exchanged.

In practice, this means that the established ISAKMP SA provides no guarantees about the intended partner of the peer. In the context of IPsec, A will next try to establish an IPsec SA using QM, which will fail, because B did not accept the ISAKMP SA. Hence this does not result in a weakness at the IPsec level. However, the IKEv1 specification [15] explicitly allows ISAKMP services to directly use the ISAKMP SA, without establishing an IPsec SA. For such services, only a weak form of authentication is guaranteed, as in (K3).

The underlying problem is that the responder's signature only includes the name of the sender (ID_B), but not that of the (supposed) peer. The signature therefore does not provide any information about who the responder believes he is communicating with. This issue can be prevented by including the name of the intended recipient inside the signature. In the above scenario, B would have inserted ID_C in step 5, and A would not have accepted the message in step 6.

(N3) Authentication failure on IKEv1 Main Mode with digital signatures that does not require self-communication. Ferguson and Schneier pointed out that the IKEv1 Main Mode with signatures is subject to a reflection attack in self-communication scenarios (K1). We find that there is another problem for the responder in this protocol, which does not involve self-communication. Critically, the third message does not include information on the initiator's intended partner. Thus, when receiving the third message, the responder cannot be sure that the message was meant for him. The attack and its consequences are similar to (N2).

(N4) Reflection attack on IKEv2 phase 2 exchange. The reflection attack that was noted before for IKEv1 Quick Mode (K2) is also possible on the related IKEv2 phase. This is surprising, because it could have been easily fixed by breaking the symmetry of the messages, e. g., by including distinct constants and checking their presence when parsing incoming messages.

Multi-protocol analysis. Our analysis of the composition of all pairs of IKE subprotocols showed that there are no problematic interferences between them. This was not a priori given as the protocols do not include explicit tagging to distinguish between the messages of different sub-protocols. The only interaction was between the "mixed" IKEv2 modes, where the sender of the last message can not be sure whether his peer is expecting (or able to parse) an authentication message of the type that he sends. However, these minor interactions clearly show that the various sub-protocols, whose joint state is defined as their long-term private keys, cannot be considered to be *universally composable* in the sense of, e. g., [7]. The full results can be found in Table 6 in Appendix B.

Compromise analysis. We analyzed each protocol with respect to the adversary compromise models described in Appendix A. The results are shown in Table 5. In the top row, the abbreviated names of the adversary models are listed.

We highlight some results. The protocol variants establish perfect forward secrecy (the **S**ecrecy sub-column in the "AF" column, where AF is the adversary model in which the adversary can compromise all long-term keys after the session) except for those without Diffie-Hellman. Interestingly, we found that some protocols revealed no attacks in the CK adversary model (which models the adversary from Canetti and Krawczyk's model for secure key exchange [5]), even when queries such as "session-state reveal" are considered, which suggests that computational proofs in CK for these protocols may be feasible. None of the considered protocols is correct in the eCK adversary model (represented by combining the columns eCK-1 and eCK-2) which also stems from the field of secure key exchange [22]. This result confirms that the IKE protocols do not offer any protection against the compromise of ephemeral keys (the short-term private Diffie-Hellman exponents). Perhaps surprisingly, many protocols seem to be vulnerable to known session-key attacks in the style of Bellare-Rogaway [3]. However, closer examination of the attacks shows that this is mainly due to

	EXT			INT			CA			AFC			AF			BR			CKw			CK			eCK-1			eCK-2		
	S	A	W	S	A	W	S	A	W	S	A	W	S	A	W	S	A	W	S	A	W	S	A	W	S	A	W	S	A	W
Signature authenticators																														
ikev1-sig-a1	○			○			○			○			○			⋆		○	⋆		○	○		○	⋆		⋆	⋆		○
ikev1-sig-a2	○			○			○			○			○			⋆		○	⋆		○	○		○	⋆		⋆	⋆		○
ikev1-sig-a-perlman1	○			○			○			○			○			⋆		○	⋆		○	○		○	⋆		⋆	⋆		○
ikev1-sig-a-perlman2	○			○			○			○			○			⋆		○	⋆		○	○		○	⋆		⋆	⋆		○
ikev1-sig-m	○			○			○			○			○			⋆		○	⋆		○	○		○	⋆		⋆	⋆		○
ikev1-sig-m-perlman	○			○			○			○			○			⋆		○	⋆		○	○		○	⋆		⋆	⋆		○
ikev2-sig																									⋆		⋆			
ikev2-sig2	○			○			○			○			○			⋆		○	⋆		○	⋆		○	⋆		⋆	⋆		○
Public key authenticators																														
ikev1-pk-a1	○			○			○			○			○			⋆		○	⋆		○	○		○	⋆	⋆	⋆	⋆		○
ikev1-pk-a12	○			○			○			○			○			⋆		○	⋆		○	○		○	⋆	⋆	⋆	⋆		○
ikev1-pk-a2																			⋆			○			⋆	⋆	⋆			
ikev1-pk-a22																			⋆			○			⋆	⋆	⋆			
ikev1-pk2-a																									⋆	⋆	⋆			
ikev1-pk2-a2																									⋆	⋆	⋆			
ikev1-pk-m	⋆			⋆			⋆			⋆			⋆			⋆		⋆	⋆		⋆	○		⋆	⋆	⋆	⋆	⋆		⋆
ikev1-pk-m2	⋆			⋆			⋆			⋆			⋆			⋆		⋆	⋆		⋆	○		⋆	⋆	⋆	⋆	⋆		⋆
ikev1-pk2-m	○			○			○			○			○					○			○			○	⋆	⋆	⋆			○
ikev1-pk2-m2	○			○			○			○			○					○			○			○	⋆	⋆	⋆			○
Pre-shared key authenticators																														
ikev1-psk-a							⋆	⋆	⋆										⋆	⋆	⋆							⋆	⋆	⋆
ikev1-psk-m	○			○			⋆	⋆	⋆	○			○					○	⋆	⋆	⋆			○			○	⋆	⋆	⋆
ikev1-psk-m-perlman	○			○			⋆	⋆	⋆	○			○					○	⋆	⋆	⋆			○			○	⋆	⋆	⋆
ikev2-mac							⋆	⋆	⋆										⋆	⋆	⋆				⋆			⋆	⋆	⋆
ikev2-mac2							⋆	⋆	⋆										⋆	⋆	⋆				⋆			⋆	⋆	⋆
Other authenticators																														
ikev2-eap																									⋆	⋆	⋆			
ikev2-eap2																									⋆	⋆	⋆			
ikev2-mactosig							⋆	⋆	⋆										⋆	⋆	⋆				⋆		⋆	⋆	⋆	⋆
ikev2-mactosig2							⋆	⋆	⋆										⋆	⋆	⋆				⋆		⋆	⋆	⋆	⋆
ikev2-sigtomac							⋆	⋆	⋆										⋆	⋆	⋆				⋆		⋆	⋆	⋆	⋆
ikev2-sigtomac2	○			○			⋆	⋆	⋆	○			○			⋆		○	⋆	⋆	⋆	⋆		○	⋆		⋆	⋆	⋆	⋆
Phase 2 subprotocols																														
ikev1-quick							⋆	⋆	⋆										⋆	⋆	⋆							⋆	⋆	⋆
ikev1-quick-nopfs	○	○		○	○		⋆	⋆	⋆	○	○		⋆	○	○	○	○		⋆	⋆	⋆	⋆	○	○	○	○		⋆	○	○
ikev1-quick-noid	○	○		○	○		⋆	⋆	⋆	○	○		○	○		○	○		⋆	⋆	⋆		○	○	○	○		⋆	○	○
ikev2-child	⋆	⋆		⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆		⋆	⋆		⋆	⋆		⋆	⋆	⋆		⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆
ikev2-child-nopfs	⋆	⋆		⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆		⋆	⋆	⋆	⋆	⋆	⋆		⋆	⋆	⋆	⋆	⋆

Table 5. Analysis results for IKEv1 and IKEv2 subprotocols with respect to different adversary models, using notation from [2] (Appendix A). Subcolumns indicate the property considered: **S**ecrecy, **A**liveness, and **W**eak agreement. Property violations that were previously reported are marked with ○, and previously unreported violations are marked with ⋆.

a technicality. In these known-key attacks, the adversary does not compute the session key from another (related) session key, but rather reveals the same session key at a so-called *non-matching session*. In our models, following [3], matching sessions are defined as sessions whose exchanged messages are matching. Consequently, known-key attacks can occur if the adversary can make an agent accept the session key while slightly modifying some messages parts. Even when these modifications do not have a practical impact on security, they lead to known-key attacks in the formal model. The relevance of such attacks is mostly theoretical; at worst, they suggest unknown-key share attacks.

Reproducing the results of the analysis. The Scyther tool, the input files, and the test scripts we used are publicly available from [10]. Although significant com-

putational effort was used to obtain the full results, only modest computational effort is required to reproduce individual attacks. For example, a single standard current-generation PC (running, e. g., Linux) would suffice to reproduce any one of the attacks described here within minutes.

5 Discussion

Our analysis reveals that many IKEv1 and IKEv2 sub-protocols do not satisfy strong security properties, such as agreement. The first underlying problem is that insufficient information about the involved agents is included in the cryptographically-protected parts of the messages. This occurs even in protocols where identity protection is not an issue. The second underlying problem is a lack of distinguishing tags in the protocols, enabling reflection attacks.

Along the same lines, though our analysis did not reveal any problematic interaction among the various sub-protocols, the lack of unique identification within the various encrypted or signed payloads remains an issue. For example, the IKE version number is only included in the header of messages and not included (or checked) within any cryptographically protected message. As a result, an active adversary can arbitrarily change version numbers and reroute encrypted content from one protocol version into another. This significantly complicates the problem of updating broken protocol versions, because the presence of the old versions may still break the security of new versions, as pointed out in [11].

In cryptographic definitions of security, such as [3], the adversary can compromise session keys of any non-matching session. As we have seen, many of the IKE variants are vulnerable to these attacks. These attacks may seem artificial; clearly, if the adversary learns the relevant session key (from whichever session) then security is lost. However, there is an important issue underlying such attacks: session keys should be computed in as few contexts as possible. The threat of computing session keys in non-matching sessions is that it may give the adversary unnecessarily more options to compromise the session key, e. g., because he can only compromise the memory of particular sessions or at certain points in time. Protocols can be designed such that the least possible amount of sessions compute the same session key, e. g., only in matching sessions, thereby minimizing the window of opportunity for compromise.

The IKEv1 attack on authentication (N2) was not reported by previous formal analysis. This is remarkable because the attack is within the scope of the methods that were used by, e. g., Meadows [25]. The reason that this attack was missed is that the property that was considered by Meadows is a much weaker form of authentication: "The receiver A of the final message should not have accepted security association SA as good for communication with B without B having itself accepted SA." [25]. It is clear from this formulation that there is no requirement stated that A's acceptance of SA for communication with B means that B accepted SA *for communication with* A. Instead, Meadows observed that the protocol does not satisfy "penultimate authentication". In our model, checking for penultimate authentication corresponds to analyzing

whether B's security guarantees also hold before the last message of the protocol is received and verified. We did not consider this property here, because we assume implementations successfully complete individual sub-protocols before assuming any security properties.

6 Conclusions

We have provided the most comprehensive formal analysis of IKEv1 and IKEv2 so far. Our analysis goes far beyond what was analyzed before using formal methods and we leveraged massive parallelization to obtain our results.

Our analysis did not reveal any new critical weaknesses in the IKE protocols with respect to the secrecy of the established keys. However, we discover several new weaknesses. These mainly revolve around the failure of the IKE protocols themselves to provide mutual authentication, even though the established session keys can be considered "safe" in the sense that they will be known, at most, to the intended partners. Hence, when using IKE, strong mutual authentication cannot be taken for granted and in fact requires an additional step where the session key is used, in such a way that reflection attacks (on exchanges protected by the session key) are prevented.

As may have been expected, IKEv2 shows significantly less weaknesses than IKEv1, and should therefore certainly be preferred over IKEv1.

The remaining weakness in IKEv2 phase 1, in which only weak authentication is achieved, in practice means that no strong authentication guarantees are provided before a message is received over the IPsec tunnel. This could be solved by either documenting the particular limitations of the signature-based Aggressive Mode, or by reintroducing the required agent field in the specification.

For IKEv2 phase 2, the practical implication of the reflection attacks is that after a phase 2 exchange, recent aliveness of the peer is not guaranteed. As above, this is resolved once a message is subsequently received over the tunnel. This could be mentioned in the specification, so that developers do not make unfounded assumptions (e. g., aliveness of the peer) after a phase 2 exchange.

With respect to future work, there are several properties that are currently out of scope of our methods. Two obvious candidates for future work are *identity protection* and resilience against *denial-of-service* attacks. Additionally, we note that in this analysis we focused on attack detection. Another possible goal is formal verification, i. e., showing the *absence* of flaws at the logical level. This seems out of scope for the current state-of-the-art in formal analysis. Along the same lines it would be desirable to prove more precise properties of the individual sub-protocols, e. g., as in [28].

References

1. Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuellar, J., Drielsma, P.H., Heám, P.C., Kouchnarenko, O., Mantovani, J., Mödersheim, S., von Oheimb, D., Rusinowitch, M., Santiago, J., Turuani, M., Viganò, L., Vigneron, L.:

The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In: Computer Aided Verification, vol. 3576, chap. 27, pp. 281–285. Springer Berlin Heidelberg (2005)
2. Basin, D., Cremers, C.J.F.: Modeling and Analyzing Security in the Presence of Compromising Adversaries. In: Computer Security - ESORICS 2010. Lecture Notes in Computer Science, vol. 6345, pp. 340–356. Springer (2010)
3. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: CRYPTO '93. pp. 232–249. Springer, New York, NY, USA (1994)
4. Boyd, C., Mathuria, A.: Protocols for Authentication and Key Establishment. Springer, 1 edn. (September 2003)
5. Canetti, R., Krawczyk, H.: Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In: EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques. pp. 453–474. Springer-Verlag, London, UK (2001)
6. Canetti, R., Krawczyk, H.: Security Analysis of IKEs Signature-based Key-Exchange Protocol. In: CRYPTO02. Lecture Notes in Computer Science, vol. 2442, pp. 143–161. Springer-Verlag (2002)
7. Canetti, R., Rabin, T.: Universal composition with joint state. CRYPTO 2003 pp. 265–281 (2003)
8. Cremers, C.J.F.: The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols. In: Computer Aided Verification. Lecture Notes in Computer Science, vol. 5123/2008, pp. 414–418. Springer (2008)
9. Cremers, C.J.F., Mauw, S.: Operational Semantics of Security Protocols. In: Scenarios: Models, Transformations and Tools, International Workshop, Dagstuhl Castle, Germany, September 7-12, 2003, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3466. Springer (2005)
10. Cremers, C., Kyburz, A.: IKEv1 and IKEv2 protocol models for the Scyther tool (2011), http://people.inf.ethz.ch/cremersc/scyther/ike
11. Cremers, C.: Feasibility of multi-protocol attacks. In: Proc. of The First International Conference on Availability, Reliability and Security (ARES). pp. 287–294. IEEE Computer Society, Vienna, Austria (April 2006)
12. Dierks, T., Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard) (Aug 2008), http://www.ietf.org/rfc/rfc5246.txt, updated by RFCs 5746, 5878
13. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions on Information Theory 29(12), 198–208 (Mar 1983)
14. Ferguson, N., Schneier, B.: A Cryptographic Evaluation of IPsec. Tech. rep., Counterpane Internet Security, Inc (2000)
15. Harkins, D., Carrel, D.: The Internet Key Exchange (IKE). RFC 2409 (Proposed Standard) (Nov 1998), http://www.ietf.org/rfc/rfc2409.txt, obsoleted by RFC 4306, updated by RFC 4109
16. Harkins, D., Kaufman, C., Kent, S., Kivinen, T., Perlman, R.: Design Rationale for IKEv2. IETF Internet Draft (expired) (Feb 2002), http://www.ietf.org/proceedings/54/I-D/draft-ietf-ipsec-ikev2-rationale-00.txt
17. Kaufman, C., Hoffman, P., Nir, Y., Eronen, P.: RFC 5996: Internet Key Exchange Protocol Version 2 (IKEv2) (Sep 2010), http://www.rfc-editor.org/info/rfc5996
18. Kelsey, J., Schneier, B., Wagner, D.: Protocol interactions and the chosen protocol attack. In: Proc. 5th International Workshop on Security Protocols. Lecture Notes in Computer Science, vol. 1361, pp. 91–104. Springer-Verlag (1997)

19. Kent, S., Seo, K.: Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard) (Dec 2005), http://www.ietf.org/rfc/rfc4301.txt
20. Krawczyk, H.: HMQV: A High-Performance Secure Diffie-Hellman Protocol. In: CRYPTO 05. Lecture Notes in Computer Science, vol. 3621, pp. 546–566. Springer-Verlag (2005)
21. Kyburz, A.: An automated formal analysis of the security of the Internet Key Exchange (IKE) protocol in the presence of compromising adversaries. Master's thesis, ETH Zurich (November 2010)
22. LaMacchia, B., Lauter, K., Mityagin, A.: Stronger Security of Authenticated Key Exchange. In: Proceedings of the 1st international conference on Provable security. pp. 1–16. ProvSec'07, Springer-Verlag, Berlin, Heidelberg (2007)
23. Lowe, G.: A Hierarchy of Authentication Specifications. In: Proc. 10th IEEE Computer Security Foundations Workshop (CSFW). pp. 31–43. IEEE Computer Society Press (1997)
24. Maughan, D., Schertler, M., Schneider, M., Turner, J.: Internet Security Association and Key Management Protocol (ISAKMP). RFC 2408 (Proposed Standard) (Nov 1998), http://www.ietf.org/rfc/rfc2408.txt, obsoleted by RFC 4306
25. Meadows, C.: Analysis of the Internet Key Exchange protocol using the NRL Protocol Analyzer. In: Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on. pp. 216–231 (1999)
26. Moedersheim, S., Drielsma, P.H., et al.: AVISPA Project Deliverable D6.2: Specification of the Problems in the High-Level Specification Language (2003), http://www.avispa-project.org/
27. Orman, H.: The Oakley Key Determination Protocol. Tech. rep., University of Arizona, Tucson, AZ, USA (1997), also described in RFC 2412.
28. Paterson, K.G., Watson, G.J.: Plaintext-dependent decryption: A formal security treatment of SSH-CTR. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 6110, pp. 345–361. Springer (2010)
29. Perlman, R., Kaufman, C.: Key exchange in IPSec: analysis of IKE. Internet Computing, IEEE 4(6), 50 –56 (Nov/Dec 2000)
30. Perlman, R.J., Kaufman, C.: Analysis of the IPSec Key Exchange standard. In: 10th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2001), 20-22 June 2001, Cambridge, MA, USA. pp. 150–156. IEEE Computer Society (2001)
31. Swiss National Computing Centre: Brutus cluster, http://www.cscs.ch/index.php
32. Zhou, J.: Further analysis of the Internet Key Exchange protocol. Computer Communications 23(17), 1606 – 1612 (2000)

A Adversary models

In this section we briefly describe the adversary models considered in this analysis as listed in Table 5. Their formal definitions can be found in [2].

EXT In this model, the adversary has full control over the network but is an outsider (EXTernal): he does not have an identity within the system and he does not know any long-term private keys.

INT The INT model (INTernal) models the standard Dolev-Yao model, as used, e. g., by Lowe for his analysis of the Needham-Schroeder protocol. The adversary has full control over the network, but he additionally has access

to some long-term private keys. This models either a malicious insider or an adversary that has compromised some agents. For example, this allows the adversary to perform Lowe's man-in-the-middle attack on the Needham-Schroeder protocol.

CA An agent A can authenticate another agent B if B knows some secrets that are not known to the adversary, such as B's long-term private keys. However, it is not necessary that A also knows some secrets. Some protocols allow for authentication even when the long-term keys of the authenticating agent (i. e., the verifier) are known to the adversary. The CA model (Compromised Actor) gives the adversary full network control but also the ability to learn the long-term keys of the authenticating agent. This adversary model is used to model Key Compromise Impersonation (KCI) attacks.

AF and AFC The AF and AFC adversary models correspond to an adversary that is capable of learning all long-term private keys of the agents after (AFter) a session, and are used to analyze Perfect Forward Secrecy and weak Perfect Forward Secrecy. Compared to the AF model, the additional restriction of the AFC model (AFter Correct) is that the adversary can only learn the keys after sessions in which he did not actively interfere, i. e., in which he was passive, which is used to model weak Perfect Forward Secrecy.

BR The BR adversary model (Bellare-Rogaway) corresponds to an INT adversary that can additionally compromise session keys of other (non-matching) sessions, thereby modeling known-key attacks.

CK and CKw The CK (Cannetti-Krawczyk) and CKw (Weak CK) models correspond to a BR adversary with two additional powers. First, the adversary can also compromise long-term keys after the session (as in AF, for CK, and as in AFC, for CKw). Second, the adversary can reveal the local state (e. g., the random numbers) generated in other sessions.

eCK-1 and eCK-2 The eCK-1 and eCK-2 models together model the eCK adversary model (Extended CK). The eCK model is similar to the combination of the CKw and CA adversary. The main difference is that instead of revealing the local state of other (i. e., non-matching) sessions, the adversary can reveal the randomness generated in every session for which he did not compromise the owner's long-term private key.

B Multi-protocol analysis results

In Table 6 we give an overview of the protocol interactions. In the left column the protocol and property considered is indicated. The markers in the top row indicate the protocols that were considered to be running in parallel. An open dot in the table denotes that a violation was found that requires self-communication. A closed dot indicates a violation that does not require self-communication. The interferences found seem to be of a harmless nature but could trivially have been avoided by putting appropriate unique constants in each message, i. e., by tagging.

No	Prot	Claim	ikev1-pk-a1	ikev1-pk-a12	ikev1-pk-a2	ikev1-pk-a22	ikev1-sig-a1	ikev1-sig-m	ikev1-sig-m-perlman	ikev2-mac	ikev2-mac2	ikev2-mactosig	ikev2-mactosig2	ikev2-sig	ikev2-sig2	ikev2-sigtomac
1	ikev1-pk-a1	I Agreement		•												
2	ikev1-pk-a1	I Weakagree		•												
3	ikev1-pk-a12	I Agreement	•													
4	ikev1-pk-a12	I Weakagree	•													
5	ikev1-pk-a2	I Agreement				•										
6	ikev1-pk-a2	I Weakagree				•										
7	ikev1-pk-a22	I Agreement			•											
8	ikev1-pk-a22	I Weakagree			•											
9	ikev1-sig-a-perlman1	R Agreement					○									
10	ikev1-sig-a-perlman1	R Weakagree					○									
11	ikev1-sig-m-perlman	R Agreement						•								
12	ikev1-sig-m-perlman	R Weakagree						•								
13	ikev2-mac	R Agreement										•				
14	ikev2-mac	R Weakagree										•				
15	ikev2-mac2	R Agreement											•			
16	ikev2-mac2	R Weakagree											•			
17	ikev2-mactosig	R Agreement								•						
18	ikev2-mactosig	R Weakagree								•						
19	ikev2-mactosig2	R Agreement									•					
20	ikev2-mactosig2	R Weakagree									•					
21	ikev2-sig	I Agreement														
22	ikev2-sig	I Weakagree														
23	ikev2-sig	R Agreement														•
24	ikev2-sig	R Weakagree														•
25	ikev2-sig2	I Agreement														
26	ikev2-sigtomac	R Agreement												•		
27	ikev2-sigtomac	R Weakagree												•		

Table 6. Multi-protocol interactions leading to property violations. These mainly involve wrong assumptions on the protocol variant that the partner is running.