Kan Zhang
Yuliang Zheng (Eds.)

LNCS 3225

Information Security

7th International Conference, ISC 2004
Palo Alto, CA, USA, September 2004
Proceedings

Springer

Kan Zhang Yuliang Zheng (Eds.)

Information Security

7th International Conference, ISC 2004
Palo Alto, CA, USA, September 27-29, 2004
Proceedings

Volume Editors

Kan Zhang
Hewlett-Packard Laboratories
3353 Alma Street, #233, Palo Alto, CA 94306, USA
E-mail: zhangkan@sbcglobal.net

Yuliang Zheng
University of Noth Carolina at Charlotte
Department of Software and Information Systems
9201 University City Blvd, Charlotte, NC 28223, USA
E-mail: yzheng@uncc.edu

Library of Congress Control Number: 2004112165

CR Subject Classification (1998): E.3, D.4.6, F.2.1, C.2, J.1, C.3, K.4.4, K.6.5

ISSN 0302-9743
ISBN 3-540-23208-7 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11325864 06/3142 5 4 3 2 1 0

Preface

The 2004 Information Security Conference was the seventh in a series that started with the Information Security Workshop in 1997. A distinct feature of this series is the wide coverage of topics with the aim of encouraging interaction between researchers in different aspects of information security. This trend continued in the program of this year's conference. The program committee received 106 submissions, from which 36 were selected for presentation. Each submission was reviewed by at least three experts in the relevant research area. We would like to thank all the authors for taking their time to prepare the submissions, and we hope that those whose papers were declined will be able to find an alternative forum for their work.

We were fortunate to have an energetic team of experts who took on the task of the program committee. Their names may be found overleaf, and we thank them warmly for their time and efforts. This team was helped by an even larger number of external reviewers who reviewed papers in their particular areas of expertise. A list of these names is also provided, which we hope is complete.

We would also like to thank the advisory committee for their advice and support. The excellent local arrangements were handled by Dirk Balfanz and Jessica Staddon. We made use of the electronic submission and reviewing software supplied by COSIC at the Katholieke Universiteit Leuven. Both the software and the ISC 2004 website were run on a server at UNC Charlotte, and were perfectly maintained by Seung-Hyun Im. We also appreciate assistance from Lawrence Teo in editing the proceedings.

September 2004

Kan Zhang
Yuliang Zheng

Information Security Conference 2004

September 27–29, 2004, Palo Alto, CA, USA

General Chair

Yuliang Zheng, University of North Carolina at Charlotte, USA

Advisory Committee

Tom Berson, Anagram Lab, USA
Li Gong, Sun Microsystems, China
Wenbo Mao, Hewlett-Packard Laboratories, UK
Eiji Okamoto, University of Tsukuba, Japan

Program Co-chairs

Kan Zhang, Hewlett-Packard Laboratories, USA
Yuliang Zheng, University of North Carolina at Charlotte, USA

Program Committee

Martin Abadi ..... UC Santa Cruz, USA
Carlisle Adams ..... University of Ottawa, Canada
Gail-Joon Ahn ..... UNC Charlotte, USA
N. Asokan ..... Nokia, Finland
Tuomas Aura ..... Microsoft Research, UK
Jean Bacon ..... Cambridge University, UK
Dirk Balfanz ..... PARC, USA
Feng Bao ..... i2r, Singapore
Elisa Bertino ..... University of Milan, Italy
Colin Boyd ..... QUT, Australia
Yvo Desmedt ..... University College London, UK
Warwick Ford ..... Verisign, USA
Craig Gentry ..... NTT DoCoMo Labs, USA
Stuart Haber ..... HP Labs, USA
Markus Jakobsson ..... RSA Labs, USA
Marc Joye ..... Gemplus, France
Michiharu Kudoh ..... IBM Tokyo, Japan
Javier Lopez ..... University of Malaga, Spain
Tsutomu Matsumoto ..... Yokohama National University, Japan
Kanta Matsuura ..... University of Tokyo, Japan
Catherine Meadows ..... Naval Research Lab, USA
Jonathan Millen ..... SRI International, USA
John Mitchell ..... Stanford University, USA
Peng Ning ..... North Carolina State University, USA
Joe Pato ..... HP Labs, USA
Josef Pieprzyk ..... Macquarie University, Australia
Jean-Jacques Quisquater ..... UCL, Belgium

Michael Reiter ..... CMU, USA
Scarlet Schwiderski-Grosche ..... Royal Holloway, University of London, UK
Hovav Shacham ..... Stanford University, USA
Dawn Song ..... CMU, USA
Jessica Staddon ..... PARC, USA
Clark Thomborson ..... University of Auckland, New Zealand
Serge Vaudenay ..... EPFL, Switzerland
Michael Waidner ..... IBM Research, Switzerland
Yumin Wang ..... Xidian University, China
Moti Yung ..... Columbia University, USA
Kan Zhang ..... HP Labs, USA
Yuliang Zheng ..... UNC Charlotte, USA
Jianying Zhou ..... i2r, Singapore

External Reviewers

Giuseppe Ateniese
Joonsang Baek
Thomas Baigneres
Julien Brouchier
Julien Cathalo
Mathieu Ciet
Scott Contini
Nora Dabbous
Chen Dan
Tanmoy Das
Alex Deacon
Anand Desai
Glenn Durfee
Dan DuVarney
Tim Ebringer
Hiroaki Etoh
Serge Fehr
Dan Forsberg
Michael J. Freedman
Steven Galbraith
Vaibhav Gowadia
Phillip Hallam-Baker
Thomas Hardjono
Zhenjie Huang
Zhengtao Jiang
Pascal Junod
Jonathan Katz
Yongdae Kim
Mei Kobayashi
Tieyan Li
Yi Lu
Benjamin Lynn
Greg Maitland
Krystian Matusiewicz
Keith Mayes
Bruce Mills
Jean Monnerat
Jose A. Montenegro
Sara Miner More
Ram Moskovitz
Zhihua Niu
Juan J. Ortega
Olivier Pereira
Gilles Piret
Zulfikar Ramzan
Louis Salvail
Diana Smetters
Mike Stay
Ron Steinfeld
Paul Syverson
Anat Talmy
Lawrence Teo
Haibo Tian
Gene Tsudik
Chenxi Wang
Guilin Wang
Huaxiong Wang
Bogdan Warinschi
Claire Whelan
Nathan Whitehead
Hao Chi Wong
Yongdong Wu
Dingbang Xu
Mariemma I. Yague
Adam Young
Ting Yu
John Zachary
Jianhong Zhang

Table of Contents

Intrusion Detection

Access Control

Human Authentication

Certificate Management

Mobile and Ad Hoc Security

Web Security

Digital Rights Management

Software Security

Practical Authenticated Key Agreement Using Passwords

Taekyoung Kwon

School of Computer Engineering, Sejong University, Seoul 143-747, Korea
tkwon@sejong.ac.kr

Abstract. Due to the low entropy of human-memorable passwords, it is not easy to conduct password authenticated key agreement in a secure manner. Though there are many protocols achieving this goal, they may require a large amount of computation specifically in the augmented model which was contrived to resist server compromise. Our contribution in this paper is two fold. First, we propose *a new practical password authenticated key agreement protocol* that is efficient and generic in the augmented model. Our scheme is considered from the practical perspective (in terms of efficiency) and is provably secure under the Diffie-Hellman intractability assumptions in the random-oracle model. Our second contribution is more realistic and generic; *a conceptually simple but novel password guessing attack* which can be mounted on every three-pass password-based protocol unless care is taken in both the design and implementation phases. This is due to the server's failure to synchronize multiple simultaneous requests. Experimental results and possible prevention methods are also discussed.

1 Introduction

User authentication is necessary for the typical case that a human being resides as a client and tries to log on to a remote server machine. The server must be able to determine the user's identity reliably over a public or private channel. Password authentication is one of such methods, in which simply the user memorizes a (short) password while the server maintains a user profile that associates the user name and the password verifying information. The intrinsic problem with this method is the memorable password, associated with each user, has low entropy, so that it is not easy to protect the password information against the notorious password guessing attacks by which attackers could search the relatively small space of human-memorable passwords.

Since a pioneering method that resists the password guessing attacks was introduced to cryptographic protocol developers [24], there has been a great deal of work for password authenticated key agreement, preceded by EKE [5], on the framework of Diffie-Hellman [10]. Readers are referred to [15] for complete references. Compared to the typical authenticated key agreement, the password-based schemes are more expensive due to the low entropy of passwords, specifically in the augmented model which was contrived to resist server compromise. Provable

K. Zhang and Y. Zheng (Eds.): ISC 2004, LNCS 3225, pp. 1–12, 2004.

security is important but tends to make the schemes harder to be practical in some cases. From the theoretical perspective, several methods that are much more expensive but provably secure in the standard model, were presented [12, 18,19]. From the practical perspective, the practice-oriented security models are applied for examining the security of protocols [1,2,3,7]. For example, EKE2 and AuthA are provably secure in both the random oracle and ideal cipher models [3, 4,8], while PAK and PAK-Z (that improves the efficiency of PAK-X impressively by specifying a generic digital signature) are in the random oracle model [7,25, 26]. However, it is (arguably) still expensive to assume ideal ciphers or digital signatures along with many costly operations on them, while PAK-Y is reasonably efficient with Schnorr signature in terms of computational costs [4,26,30].

At present, SPEKE [16], SRP [32], PAK [26], and AMP [21] are being discussed by the IEEE P1363 Standard Working Group and more recently by the ISO/IEC JTC 1/SC 27 group as practical protocols for standardization on password-based public key cryptographic techniques [13,14]. Among them, PAK and SPEKE are 'three-pass' protocols, while AMP and SRP are 'four-pass' protocols. The standardization work is valuable in many aspects; for instance, a new attack called the 'two-for-one' guessing attack[1] against the four-pass protocols was found and resolved in the process [13,31]. Any preference between three-pass and four-pass is still open for password-based protocols while typical authenticated key agreement such as STS and SIGMA is three-pass [11,20].

In this paper, our contribution is two fold from the practical perspective.

1) An efficient three-pass password-based protocol in the augmented model
2) A generic password-guessing attack against three-pass protocols

A password-based protocol designed in the augmented model can resist server compromise. In other words, an adversary who compromised a password profile from a server cannot impersonate a user without launching dictionary attacks. For this additional property, the related protocols (for example, A-EKE, AMP, AuthA, B-SPEKE, PAK-Z, and SRP) are more expensive than those are not (for instance, EKE, EKE-2, SPEKE, and PAK) in the augmented model [6,21,4,17, 26,32]. We observe that the existing provably-secure schemes are still expensive in the augmented model in terms of the amount of computation, and that it is desirable to minimize the number of message passes and the size of message blocks for practice on expensive communication channels. So we design a new three-pass password-based protocol in the augmented model with both security and efficiency in mind. We achieve this goal interestingly by a composition under the careful observation of the existing schemes discussed by the IEEE P1363 Standard Working Group, say without losing the presumed level of security. We call the protocol TP-AMP and prove its security in the random oracle model.

On developing the new three-pass password-based protocol, we find a conceptually simple but novel password guessing attack which can be mounted on every

[1] An active attacker can validate two password guesses in one impersonation attempt. The first attack against SRP was discovered by D. Bleichenbacher in 2000, while the similar attack on AMP was by M. Scott [31]. However, both protocols were fixed to resist respective attacks by each original author [13].

three-pass password-based protocol by exploiting a small window of vulnerability resulting from a standard technique to resist on-line guessing attacks, say from counting the number of failed requests. Our attack is due to the server's failure to synchronize multiple simultaneous requests, and is unavoidable in three-pass protocols unless special care is taken in both the design and implementation phases. We call this attack a *many-to-many* (or *parallel*) guessing attack[2] because an active attacker can validate as many password guesses as (s)he makes server instances invoked concurrently, regardless of its upper limit of on-line guessing. A prototype of the proposed protocol is implemented to show how our attack works and is prevented. We first consider this attack and possible resolution in the literature.

This paper is organized as follows. In the following section, the so-called TP-AMP protocol (our first contribution) is presented. In Section 3, the many-to-many guessing attack (our second contribution) is described in more detail. In Section 4, security and efficiency of TP-AMP are discussed. Finally this paper is concluded in Section 5.

2 A Practical Protocol

2.1 Preliminaries

Our principal motivation comes from the fact that password-based protocols designed in the augmented model are much less efficient than those are not in that model, in terms of either computation or communication costs. When we regard PAK as a fundamental structure for three-pass protocols due to its simplicity and clarity, we can easily observe that its augmentation such as PAK-X, PAK-Y, and PAK-Z are far from its intrinsic nature and get much more complicated in the augmented model [7,25,26]. AMP and SRP show better performance in that model but in four passes [21,32]. So, our basic idea is to make AMP squeezed into PAK or PAK augmented by AMP, since AMP is another protocol that can be computed very efficiently over various numerical groups [21]. However, a simple composition is not sufficient, and consequently we obtain a new practical protocol by more careful consideration on them.

The reason for choosing PAK rather than EKE2 is obviously that the former can formally be proved by postulating the random oracles only, while the latter requires the additional assumption of ideal cipher [7,26,3]. However, EKE2 or similar schemes that are proved sufficiently secure, can also be applied to constructing the practical augmented protocol in the way of our composition. In that sense, our construction is quite generic.

In Table 1, we enumerate the notation, in part, to be used in the remaining of this paper. Additional ones will be self-contained in each part of this paper. Let κ be a general security parameter (say 160 bits) and ℓ be a special security parameter for public keys (1024 or 2048 bits). A client C and a server S should

[2] We first introduced this attack at IEEE P1363.2 meeting and also discussed a few names for it.

Table 1. Basic Notation

C	Client (User)	S	Server
π	Password	τ_C	Transformed password for C
$\leftarrow$	Derivation	$\stackrel{R}{\leftarrow}$	Random selection
κ, ℓ	Security parameters	q	Prime of size κ
r	Integer co-prime to q	p	Prime of size ℓ such that $p = rq + 1$
$\mathbb{Z}_p^*$	Multiplicative group of p	$\mathbb{G}_q$	q-order subgroup of $\mathbb{Z}_p^*$
g	Generator of $\mathbb{G}_q$	h_i, H_i	Random oracles
α, β	Agreed values	sk_i	Session key

agree on algebraic parameters[3] related to Diffie-Hellman key agreement such as p, q, and g. Define $\bar{\mathbb{G}}_q = \{g^x \bmod p | x \in \mathbb{Z}_q^*\}$ where $|\bar{\mathbb{G}}_q| = q-1$. Let us often omit 'mod p' from the expressions that are obvious in $\mathbb{Z}_p^*$. Let $\{0,1\}^*$ denote the set of finite binary strings and $\{0,1\}^n$ the set of binary strings of length n. We then define random oracles such that h_i: $\{0,1\}^* \rightarrow \{0,1\}^\kappa$ and H_i: $\{0,1\}^* \rightarrow \{0,1\}^\ell$. Their instances are also defined as $h_i(\cdot) = h(i, \cdot, i)$ and $H_i(\cdot) = (h(i, \cdot, i))^{\frac{p-1}{q}}$ mod p where $h(\cdot)$ is a strong one-way hash function. Let ACCEPTABLE$(\cdot)$ denote an acceptable function which may return true if its pre-image satisfies the given security properties, as defined in Section 2.3. Readers who are not familiar with the legacy protocols, are referred to the previous work of [7,13,21,25,26].

2.2 Proposed Protocol – TP-AMP

TP-AMP stands for the Three-Pass Authenticated key agreement via Memorable Passwords and is depicted in Figure 1. Let us borrow the name AMP from [21] for our basic motivation.

Protocol Setup. On the registration phase, a user chooses a name C and a password π while the server S saves user's profile $\langle C, \tau_C \rangle$ in its stable storage where $\gamma = H_0(C, \pi)$, $\gamma' = \gamma^{-1} \bmod p$, $u = h_1(C, \pi)$, $\nu = g^u$, and $\tau_C = \langle \gamma', \nu \rangle$. For convenience, S is assumed as an IP address of the server machine.

Protocol Run. A user may type C and π into the client machine. The client (C on behalf of the user from now on) then chooses x at random from $\mathbb{Z}_q^*$ (not $\mathbb{Z}_p^*$), and computes γ in order to obtain $m = g^x\gamma$. The client sends ($\rightarrow$) a commitment message $\langle C, m \rangle$ to the server.

$$1.\ C \rightarrow S : C, g^x\gamma$$

[3] In spite that PAK, in general, does not require gcd(r, q)=1 and only PAK-R requires it for further randomization, we recommend to use a *secure prime* such that each factor of r except 2 is of size at least κ or a *safe prime* such that $r = 2$ for $p = rq+1$ as discussed in [21,23,32,29]. They satisfy gcd(r, q)=1. Specifically, we observe that TP-AMP shows the best performance with a secure prime, while PAK-Y does with a safe prime and arbitrarily smaller exponents [28].

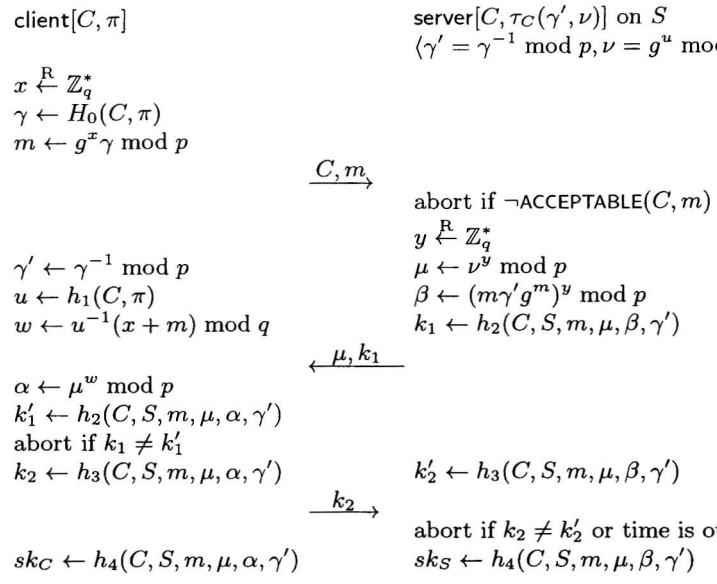


Fig. 1. TP-AMP (Three-Pass AMP Protocol)

After or before sending message 1, the client could compute γ' and the user's *amplified password* such that $w = u^{-1}(x + m) \bmod q$ by obtaining $u = h_1(C, \pi)$, and keeps them while waiting for message 2. In practice, we can hash m so that we have $q|h(m)$ with negligible probability.

Upon receiving message 1, the server should abort it if ACCEPTABLE (C, m) returns false. Otherwise, the server fetches $\langle C, \tau_C \rangle$ from its storage and chooses y at random from $\mathbb{Z}_q^*$ so as to obtain $\mu = \nu^y$. The server then computes $\beta \equiv (m\gamma' g^m)^y \equiv g^{(x+m)y} (\bmod\ p)$ and $k_1 = h_2(C, S, m, \mu, \beta, \gamma')$, and sends a challenge message $\langle \mu, k_1 \rangle$ to the client.

$$2.\ S \rightarrow C : \nu^y, h_2(C, S, m, \mu, \beta, \gamma')$$

After or before sending message 2, the server could compute $k_2' \leftarrow h_3\ (C, S,\ m,\ \mu,\ \beta,\ \gamma')$ and keeps it while waiting for message 3. The server should abort if time is run out.

Upon receiving message 2, the client raises μ to the amplified password so that $\alpha \equiv \mu^w \equiv g^{y(x+m)}\ (\bmod\ p)$, and computes $k_1' = h_2(C, S, m, \mu, \alpha, \gamma')$. If k_1 is not equal to k_1', the client should abort this session. Otherwise, the client computes $k_2 = h_3(C, S, m, \mu, \beta, \gamma')$ and sends a response message k_2 to the server.

$$3.\ C \rightarrow S : h_3(C, S, m, \mu, \alpha, \gamma')$$

After or before sending message 3, the client could compute a session key such that $sk_C = h_4(C, S, m, \mu, \alpha, \gamma')$ and deletes any other ephemeral values.

Upon receiving message 3, the server should abort this session if k_2 is not equal to k_2'. Otherwise, the server should compute a session key such that $sk_S = h_4(C, S, m, \mu, \beta, \gamma')$ and deletes any other ephemeral values.

As a result, the client and the server could authenticate each other using the passwords and agree on the same session key $sk_C (= sk_S)$ because $\alpha \equiv \beta \equiv g^{(x+m)y} (\mathrm{mod} p)$.

2.3 Small Discussion

One can easily see that message 1 is extracted from PAK while message 2 and session key are motivated by AMP. This protocol performs simple computation in three passes and works in the augmented model where τ_C is defined as $\langle \gamma', \nu \rangle$. For efficiency, it would be better to hash m when we compute β and w, say $\beta = (m\gamma' g^{h(m)})^y$ and $w = u^{-1}(x + h(m)) \bmod q$ for a strong one-way hash function $h(\cdot)$. For more efficiency, we recommend to use a secure prime for TP-AMP. Security and efficiency of the proposed protocol will be discussed in Section 4.

In the legitimate protocol run, g^x and ν^y are assumed not to be trivial values such as 0 and 1 as in the Diffie-Hellman relatives. We need to define a *failure count* that must be manipulated by the server and increased by one when $k_2 \neq k_2'$. The server should abort further requests of the client if the (subsequent) failure count exceeds its pre-defined limit, δ. This is a standard technique for resisting on-line guessing attacks. We also need to define the special function called ACCEPTABLE$(\cdot)$ since the server should abort when it returns false upon receiving $\langle C, m \rangle$. An example of the function follows:

ACCEPTABLE$(\cdot)$

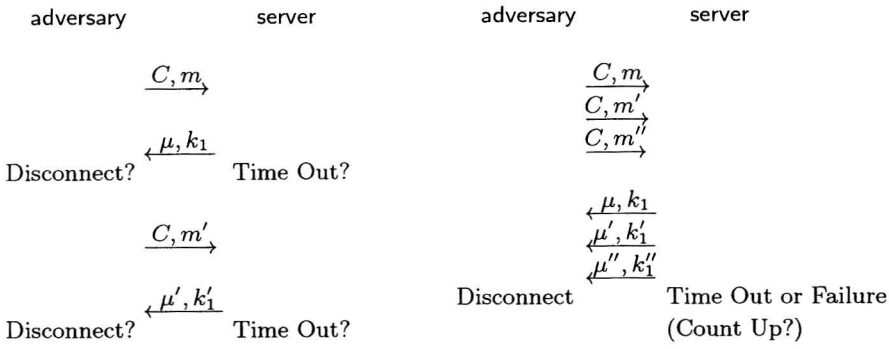
INPUT: $\langle C, m \rangle$
OUTPUT:
Return *false*
  if C **is being served by another instance; /* See Section 3 */**
  else if the failure count of C **is greater than or equal to its limit** δ**;**
  else if $q|m$**; /* Check if** $m \notin Z_p^*$ **only when hashing** m **before raising** g ***/**
Return *true* **otherwise;**

Note that the first condition (for resisting the many-to-many guessing attacks in the next section) can be considered in very flexible ways, for example, an IP address instead of C, and can be substituted by a more effective way in the future. This function is valid for authentication sessions only. Note also that $q|m$ means q divides m, but it might be enough to assure $m \in Z_p^*$ only when we hash m for β and w in the protocol.

3 Many-to-Many Guessing

3.1 A Real World Attack

It is widely recognized that three-pass (say, smaller-pass) protocols are favorable to the channel efficiency for authenticated key agreement. However, care must be taken for password authenticated key agreement in a practical sense.



(a) Sequential Attempts
- Exercise -

(b) Parallel Attempts
- Real Attack -

Fig. 2. Basic Concept of Many-to-Many Guessing Attacks

Let us glance over Theorem 1, in advance, that is introduced in Section 4 and proved in [22]. There exists an adversarial advantage that is bounded by $\frac{q_{se}}{N}$. The similar results can be found from the closely related work [3,8,26]. These advantages imply that the adversary is reduced to a simple online guessing attacker that can easily be detected and prevented from exceeding the pre-defined limit, δ, on the number of sequential on-line trials allowed by the server's policy. For example, an adversary posing as a user C sends an arbitrary message $\langle C, m \rangle$ to the server, based on her guessed password. The server may respond with $\langle \mu, k_1 \rangle$ in the three-pass protocols while only μ in the four-pass protocols. Then, the adversary is assumed to check her guess with probability bounded by $\frac{q_{se}}{N}$ under the limit δ in three-pass protocols. Is this standard assumption really true?

Unfortunately, the answer is No! This classical prevention method can be fooled out of making the adversarial advantage much larger and in some cases disclosing a password, in a surprisingly simple way. Figure 2 depicts the possible bad events. Our attack is motivated from the fact that the server is typically implemented as a multi-threaded or multi-process application for handling many user requests simultaneously, and that the three-pass password-based protocol is not an exception. As summarized in Figure 2-(a), the adversary is able to exercise the real attack (that is described in Figure 2-(b)), for example, in order to approximate the maximum amount of time the server may wait for the third message k_2. The adversary then starts simultaneous authentication sessions, which the server processes independently in separate threads, and in that amount of time, is able to drive many different initiating messages based on different password guesses concurrently to the server. The adversary may get as many replies as allowed in that time boundary, by exceeding δ obviously. Figure 2-(b) abbreviates this idea. It could be a real world attack from the automated (and multi-threaded) adversary. The server instances must respond to each request and wait for the replies k_2 from the adversary who can even dis-