Bellcore
Bell Communications Research

Memorandum Abstract

Memorandum No. (IM or TM) TM-24531	Six Character External Project Number(s) 44ARAM	Memorandum Completion Date October 6, 1994

Title

A secure joint signature and key exchange system

Software/Product Name	Release No.

Contact/SME(s)	Org. Code(s)	Loc. Code & Room No.(s)	Tel. No.(s)
Ravi Ganesan (Bell Atlantic)	Silver Spring, MD 20904		(301) 236-7583
Yacov Yacobi	21533	MRE 2Q-338	(201) 829-4668

Proprietary Status

- □ Bellcore Proprietary - Internal Use Only
- □ Bellcore and (Listed Entities) Proprietary - Internal Use Only
- □ Bellcore Confidential - Restricted Access
- □ Bellcore and (Listed Entities) Confidential - Restricted Access
- □ Bellcore Confidential - Addressee Only
- □ Bellcore and (Listed Entities) Confidential - Addressee Only
- ☒ Non-Proprietary

Licensed Status □ Licensed Material - Property of Bellcore

Listed Entities - Information Also Proprietary/Confidential To:

□ Ameritech	□ NYNEX	□ U S WEST	□ Other
□ Bell Atlantic	□ Pacific Bell	□ SNET	
□ BellSouth	□ Southwestern Bell	□ CBI	

Entitled Companies

☒ Ameritech	☒ NYNEX	☒ U S WEST	□ Other
☒ Bell Atlantic	☒ Pacific Bell	☒ SNET	
☒ BellSouth	☒ Southwestern Bell	☒ CBI	

Subsidiaries Not Entitled

Abstract (Abstract Text, Author Signature(s), Copy To Information)

Boyd [3] proposed a simple useful extension of RSA for digital multi-signatures. In the extension, the RSA private exponent is split into multiple portions, which can then be used by multiple parties to create a joint signature.

In this work we begin by analyzing the security of the system (Boyd does not perform such an analysis) and proving some interesting properties. Next, importantly for practical applications, we show that many security properties are retained even when the private exponents are relatively short (unlike in ordinary RSA). Finally we show how the system can be used to establish secure channels, and give evidence of security.

An important practical application, is when a central server maintains a share of every user's private key (such that the user does not know that share), and users sign messages jointly with the server. A digital signature infrastructure which has such a central server provides for a convenient, non by-passable, audit point, and permits instant key-revocation. If desired, it is possible to make the user's portion of the private key short (say 64 bits), which is very useful in an era when smart cards (and smart card readers) are not ubiquitous on computing/telecommunications equipment.

Ravi Ganesan

Yacov Yacobi Yacov Yacobi

1 Introduction

1.1 Overview

A joint signature system is one in which two (or more) parties must collaborate in order to compute the digital signature – no single party can compute such a signature independently. Several such schemes have been proposed, and the scheme we focus on was originally developed by Boyd[3] and subsequently re-invented by us. The system works by splitting, in a particular manner, the RSA private exponent into multiple parts, with each part entrusted to a different entity. To sign a message, each entity, in turn, performs the regular RSA operation of modular exponentiation. Upon completion, the resulting signature is indistinguishable from one created with the original exponent which was split.

Our primary focus is on the case when the RSA private exponent is split into two portions, though we conjecture that several of our results generalize to the multiple entity case. Unless otherwise stated, from now we refer specifically to the two entity case.

This paper makes three contributions:

- We prove that the system has strong security properties which guarantee that cracking it is equivalent to cracking RSA.
- Next we prove that several of these security properties are retained even when one of the two portions is relatively short (e.g. 64 bits instead of 512 bits).
- Finally, we describe, and prove security properties of a key exchange protocol, based on the previous building blocks.

A practical application of our system, addresses the case when one of the two entities is a central server, with which a user (the other entity) must interact in order to compute a signature. The presence of such a server has several important practical benefits including provision for short user keys, a central point for auditing, and instant key revocation, and last, but not least, it provides a method of digital signature, with short secret keys, that users can memorize. This last property is valuable currently, since smart cards and smart card readers are not yet ubiquitously available on communications and computation equipment.

1.2 Other possible solutions to the short key problem

One alternative solution to the problem of long private keys is to store the private key on a server, encrypted with a conventional cryptosystem, such as DES. Upon request, the server gives the user the encrypted private key, which the user decrypts using the DES key, which is typically implemented in the form of a password known only to the user. Our scheme compares favorably to this method from a security perspective, since in our scheme an eavesdropper will not be able to mount off-line password guessing attack.

Why not just let the regular RSA private key be short? Wiener's result[10] implies that this would require that very long public keys are used (1.5 times the modulus length). Further, in such a system joint signatures are just two independent signatures, and verification is thus two independent verifications (at least one of which must be with the very long public key). In contrast, in our solution a user has a short secret key, and verification of the joint signature is equivalent to one ordinary RSA verification (and we do not need to trust the server).

1.3 Other related work

In [5] Desmedt and Frankel proposed a (t, n) threshold signature scheme where any t out of n pre-designated parties can jointly sign for the whole set of n parties. The system we describe is similar (but not identical) to one of the DF schemes, with $(t, n) = (2, 2)$. However, unlike our system, the DF system needs a clearinghouse, to combine the partial results, and unlike the DF system, the system we describe can be used to establish a secure channel.

In [8] Micali proposes a public key system, such that the shares of each user's secret key are given to n escrow authorities, every t of which can jointly reconstruct the whole secret key (upon warrant). Revealing the user's private key in this fashion is dis-advantageous in that the same private key cannot be used as the basis for a digital signature system. Further, it is no longer possible to provide eavesdropping on selective session, since the authority has the power to eavesdrop on all subsequent sessions. In contrast, if our secure channel protocol is used with a central security server and two users, i_1 and i_2, then the security server can mediate a key agreement between i_1 and i_2, after which j, the server, knows the session key. Under warrant j will divulge such (i_1, i_2) session key to the authorities, without exposing user's permanent secret keys (which are not known to the server). So, our scheme allows more penetration (of a trusted authority) into session keys, but less invasion into permanent secret keys. We believe that in some contexts this balance makes sense.

Bellovin and Merritt [1] create a secure $i - j$ channel, starting from a short secret password, known to both i and j, hence, unlike in our system, j must be fully trusted. Also, they do not create signatures.

1.4 Outline of the paper

In section 2 we describe the signature protocol, and in section 3 we show how the same mechanisms could be used to establish secure channel between the signer and the server. Each of these sections is equipped with evidence of security for the corresponding protocols. Section 4 concludes the paper.

2 Joint Signatures and Proof Of Security

2.1 Key Generation

Let i and j be two parties that want to jointly sign message m. Let v denote any verifier. We extend the RSA public-key cryptosystem as follows: let the joint secret exponent of i and j be d_{ij}, the corresponding joint public exponent e_{ij}, the joint modulus $N_{ij} = p \cdot q$, p, q are large properly chosen primes, and $d_{ij} \cdot e_{ij} \equiv 1 \bmod \lambda(N)$, where $\lambda(N)$ denotes the Carmichael function of N. Unlike in ordinary RSA, none of i or j knows the factorization of N, nor any related function, such as $\phi(N), \lambda(N)$. Further, neither i nor j know d_{ij}. All i knows besides the public information is some d_i, and similarly j knows a d_j, such that $d_i \cdot d_j \equiv d_{ij} \bmod \lambda(N)$. Some certification authority is the only one who (for a short while) knows p and q, while creating the above keys. This factorization is destroyed thereafter.

As we shall see later, it is possible to use this protocol with a relatively short secret exponent, d_i. If used in this fashion, we impose an additional size constraint on d_i. Let $n = log(N)$ be the security parameter. Certainly d_i must be super-polynomial in n. Any "small" superpolynomial function in n is good enough, for example, we can use $d_i = O(F(n))$, where $F(n) = n^{log(n)/loglog(n)}$.

Signature Protocol:

To jointly sign a message $m \in Z_N$ i computes $c_i \equiv m^{d_i} \bmod N$, sends it to j, who verifies that indeed $c_i^{d_j \cdot e_{ij}} \equiv m \bmod N$, and if so sends to v $c_j \equiv c_i^{d_j} \bmod N$. The joint signature is c_j. The verifier, v checks to see whether $m \equiv c_j^{e_{ij}} \bmod N$.

It follows from theorem 5.1 in [4] that if i and j collude they can factor N. We summarize the protocol in the following table (omitting verification processes).

Signature Protocol

i		j
$c_i \equiv m^{d_i}$	$\longrightarrow$	
	$\longleftarrow$	$c_j \equiv c_i^{d_j}$
Send verified c_j to verifier, v.		
	□	

2.2 Evidence of security

2.2.1 Passive adversary

The signature forgery problem for passive adversary is (for simplicity we use e and d to denote e_{ij} and d_{ij}, respectively):
Problem A:
Input: N, e, m_0, and a history of polynomially many triples $\{m, m^{d_i}, m^d\}$,
Output: c_0, s.t. $c_0^e \equiv m_0 \bmod N$.

The RSA forgery problem is:
Problem B:
Input:$\mathcal{N}$, E, M_0,
Output: C, s.t. $C^E \equiv M_0 \bmod \mathcal{N}$.

Theorem 1: There exists a randomized Turing reduction [2] from problem **B** to problem **A**.
Sketch of Proof: Map $\mathcal{N} \rightarrow N, E \rightarrow e, M_0 \rightarrow m_0$, and pick d_i with homogeneous distribution from its specified domain (of short exponents). It follows that $D = d$, and $M_0^D \equiv m_0^d \bmod N$. To create a consistent history $\{m, m^{d_i}, m^d\}$ proceed as follows: Pick with homogeneous distribution[1] numbers c, which will play the role of $m^d \bmod N$, and compute $m \equiv c^e \bmod N$. Then compute $m^{d_i} \bmod N$. Clearly, the output of oracle **A** is the answer for problem **B**. Assuming that the natural distributions of problems **A** and **B** are homogeneous, and by the randomization on d_i we conclude that the domination property holds [2]. Q.E.D.

2.2.2 Active adversary

RSA has its security problems, which stem from its multiplicative property, namely, given the signatures σ_1, σ_2 of some user on two messages, m_1, m_2 one could forge his signature on $m_1 \cdot m_2 \bmod N$ (it is $\sigma_1 \cdot \sigma_2 \bmod N$). In practice we overcome this problem by imposing some structure on the messages to be signed. For example, a message may have to be bounded to the left by $0^k, 1$ and by $1, 0^k$ to the right, where k is linear in the security

[1] Or any desired distribution of history (provided it is P-samplable [2]). We assume henceforth homogeneous distribution.

parameter. It is highly unlikely that multiplicative properties can still be efficiently exploited with this structure. The same restrictions and assumptions must be used in our system.

The formalization of impersonation attack by some z is that z plays in the middle, and instead of transferring c_j (say), he transfers $h(c_j)$ for some efficiently computable function $h()$, such that the protocol is not aborted, i.e. all verifications conclude positively, for some message $\tilde{m}_0 \not\equiv m_0 \bmod N$. We make the assumption that for properly structured messages this is impossible. Let $\mathcal{M}$ denote the domain of properly structured messages. Throughout "efficient" means "computable in probabilistic polynomial time on the average."

Assumption-1: There is no efficient algorithm that can forge RSA signature on non negligible fraction of $\mathcal{M}$.

Since each of c_i and c_j is an ordinary RSA signature for the legitimate recipient, assumption-1 implies that $h()$ must be the identity function, hence there is no difference between active and passive adversary, and the proof of the previous section applies.

2.2.3 Attacks by i against j

In addition to the input of problem **A** if the attacker is i, then he has d_i. The reduction from problem **B** to the new problem **A'** still holds. This works regardless of the length of d_i.

To analyze how hard it is for j to generate a joint signature we analyze the symmetrical problem. However, now the reduction works only if d_i is full size (since after picking d_j, and given d, it is most likely that the corresponding d_i is full size). We conclude:

Theorem 2: (i) It is as hard for i to create a joint $(i-j)$ signature by himself as to forge ordinary RSA signature, and
(ii) It is as hard for j to create a joint $(i-j)$ signature by himself as to forge ordinary RSA signature, provided that d_i is full size.

Corollary:
It follows from Theorem 2, that i cannot forge j's signature $m_0^{d_j}$ under assumption-1, since such a forgery implies the ability to forge a joint signature. Similarly j cannot forge i's signatures, $m_0^{d_i}$ under assumption-1, if d_i is full size (and probably even if d_i is shorter, but that was not proved).

Wiener's Attack In [10] Michael Wiener showed how to find in polynomial time a secret RSA exponent, given only the public information. His attack works if

1. The secret exponent is of length less than $\approx |N|/4$.
2. The public exponent is of length less than $\approx 3|N|/2$.

For an outsider, z, as well as for i (acting against j) the first requirement does not hold. For j (acting against i) the second requirement does not hold; the effective public exponent that j has, with respect to i's secret key, d_i, is $e \cdot d_j$. As we mentioned before, we do not know if the added information that j has, namely the factorization of $e \cdot d_j$ into large factors e and d_j, can help him.

3 Secure i-j channel

To establish a secure channel between i and j each of these parties picks a number (x for i and y for j) with homogeneous distribution from $(1, N)$, raises it to the power of his secret exponent, and sends to his counterpart, who deciphers it by raising it to his secret exponent, and then to the joint public key. The session key is $sk \equiv x \cdot y \bmod N$. Then the parties try the session key using two way challenge response with a strong conventional cryptosystem. If the process concludes positively they have authenticated each other, and agreed on a session key. The process is summarized (omitting details) as follows:

	Secure Channel	
i		j
pick x		pick y
x^{d_i}	$\longrightarrow$	
	$\longleftarrow$	y^{d_j}
	$sk \equiv x \cdot y \bmod N$	
	$\Box$	

We show that breaking this protocol is as hard as breaking RSA for properly structured messages, where the public key is short. Interestingly, in this reduction the short RSA public key is mapped into the short secret key of i in this protocol. As before, we assume that $h() = I$ the identity function, and concentrate on passive attack. We use **B'** to denote RSA cracking problem, which is like problem **B**, with the added assumption that E is relatively small (but super-poly, say O(F(n))). We use **C** to denote the problem of cracking the above protocol.

Problem C:
Input: $a = x^{d_i}, b = y^{d_j}, N, e$ and a polynomial history $\{a', b', sk'\}$,
Output: $sk \equiv x \cdot y \bmod N$.

Theorem 3: There exists a randomized Turing reduction from **B'** to **C**.
Sketch of Proof: Map $\mathcal{N} \rightarrow N$, $E \rightarrow d_i$ (they are both short), $M_0 \rightarrow a$, and pick e and b with homogeneous distribution from the appropriate domains. Compute $y \equiv b^{d_i \cdot e} \bmod N$. Create a consistent history $\{a', b', sk'\}$ as follows: Pick x' and compute $a' \equiv x'^{d_i} \bmod N$. Pick b' and compute $y' \equiv b'^{d_i \cdot e} \bmod N$. Compute $sk' = x' \cdot y' \bmod N$. From oracle C's output $sk \equiv x \cdot y \bmod N$ extract $x \equiv sk \cdot y^{-1} \bmod N$. Clearly $x \equiv C \bmod N$. The domination property holds [2]. Q.E.D.

Note: A verifier of a signature by user i on message m could conduct a pass-word guessing attack on the (i, j) channel, to recover the relatively short secret key d_i, unless this channel is protected, e.g. using our secure channel protocol. An alternative solution is proposed in [6].

4 Conclusions

Under assumption-1 both the signature and secure channel protocol are protected against outsider z, and i and j are protected against each other, for $m_0, x, y \in \mathcal{M}$, assuming that user i uses full size secret key. In the case that user i uses short secret key the above proofs extend for protection against outsiders, and protection of j against i. We believe that in addition users (i) are protected against the server, j, but did not prove it. We leave it as an open problem.

References

[1] S. Bellovin and M. Merritt, *Encrypted Key Exchange: Password Based Protocols Secure Against Dictionary Attacks*, Proc. IEEE Computer Society Symposium on Research in Security and Privacy , 1992, pp. 72-84.

[2] M. Ben-Or, B. Chor, O. Goldreich, Levin, *On the Theory of Average Case Complexity*, STOC'89.

[3] C. Boyd, *Digital Multisignatures*, Cryptography and Coding, Clarendon Press, Oxford 1989, H.J. Beker, and F.C. Piper, Ed.

[4] E. Kranakis *Primality and Cryptography*, Wiley & Sons, 1986.

[5] Y. Desmedt, and Y. Frankel, *Shared Generation of Authenticators and Signatures*, Proc. Crypto'91, LNCS 576, pp. 457-469. Springer-Verlag, 1992.

[6] Ganesan, R., *Yaksha: Kerberos + RSA*, (submitted for publication)

[7] A.K. Lenstra and H.W. Lenstra, *Algorithms in Number Theory*, The University of Chicago, TR 87-008, May 1987.

[8] S. Micali *Fair public key*, Proc. Crypto'92.

[9] R. Rivest, A. Shamir, and L. Adleman, *A Method of Obtaining Digital Signatures and Public Key Cryptosystems*, CACM, Vol. 21, pp 120-126, February 1978.

[10] M.J. Wiener, *Cryptanalysis of Short RSA Secret Exponents*, IEEE Trans. on IT, May 1990, Vol. 36, No. 3, pp553-558.

[11] R. Yahalom, B. Klein, and T. Beth, *Trust Relationship in Secure Systems – A Distributed Authentication Perspective*, Proc. of 1993 IEEE Computer Society Symposium on Research in Security and Privacy.

Yacov Yacobi

Ravi Ganesan
Senior Manager
Center for Excellence for Electronic Commerce, Bell Atlantic

Yacov Yacobi
MTS
Network Security Research

Copy to
215 Directors and Exec. Dirs.
All members of Departments 21530
A. V. Aho
J. Bellisio
J.E. berthold
D. Duffy
K. Fischer, Bell-Atlantic
R.F. Graveman
J.N. Giacopelli
J. Kimmins
H. Kluepfel
K.J. Lutz
R.W. Lucky
W. Parran, Bell-Atlantic
R. Pyle, Bell-Atlantic
J.F. Rizzo
R. Shank

H. Sherry
N. Sollenberger
E.W. Soueid
B. Stump, Bell-Atlantic
M. Wegleitner, Bell-Atlantic
R. Wolff
Abstracts only to:
W.J. Barr
G.H. Heilmeier
S.D. Personick
B.K. Schwartz
J.C. Zolnowski