社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ＤＰＳ－117　(31)
2004－CSEC－　24　(31)
2004／3／5

公開鍵利用プロトコルに対するMITM攻撃検出手法およびそのSSHへの適用

稲村 雄 †　　本郷 節之 †

† 株式会社 NTT ドコモ マルチメディア研究所
〒 239-8536 神奈川県横須賀市光の丘 3-5
NTT DoCoMo R&D センタ

{jane,hongo}@mml.yrp.nttdocomo.co.jp

あらまし　公開鍵暗号技術をナイーブに適用したプロトコルには MITM ({Man,MIG,Monkey}-In-The-Middle) 攻撃 (通信者の間に位置する攻撃者によるデータ改竄・盗聴攻撃) に弱いという弱点がある．そのような MITM 攻撃の実行を PKI 等の大掛かりな枠組に頼ることなく検出する手法と，同手法をインターネットセキュアプロトコルの一種である SSH に適用した結果を報告する．

A Detection Method for MITM Attack against Public-key Protocols and Its Appllication to SSH

Yu Inamura†　　Sadayuki Hongo†

†NTT DoCoMo, Inc. Multimedia Laboratories
3-5, Hikarinooka, Yokosuka
Kanagawa, 239-8536 Japan

{jane,hongo}@mml.yrp.nttdocomo.co.jp

Abstract　It is well known that the protocols to which public-key cryptography is naively applied are vulnerable to the so-called "{*Man,MIG,Monkey*}*-In-The-Middle (MITM)*" attack, which is conducted by an adversary sitting between the sender and the receiver and mediating transfered data with modification at her/his will. We present a neat method to counter such MITM attacks without relying upon the possibly overkilling frameworks such as PKI. The experimental results with its application to an Internet secure protocol SSH is also exhibited.

1　はじめに

近年，携帯電話/PDA 等の計算資源が限られたプラットフォームでも，公開鍵暗号利用プロトコルは一般的なものとなりつつある．現在では主として SSL を用いたセキュアウェブアクセスというのが主たる利用方法ではあるが，今後は PC 同様多種類の公開鍵プロトコルが利用されるようになることは容易に推測できる．そのような通信環境の実現に先立って，十分安全な利用技術を研究しておく必要があるだろう．さて，共通鍵暗号技術の根本的欠点を解決する手段として考案された公開鍵暗号 [1] では，ある受信者 A に対して秘密裏にメッセージを送るために必要となるのは A の公開鍵でメッセージを暗号化することのみであり，A 自身だけが知る秘密鍵を用いない限り当該メッセージを復号す

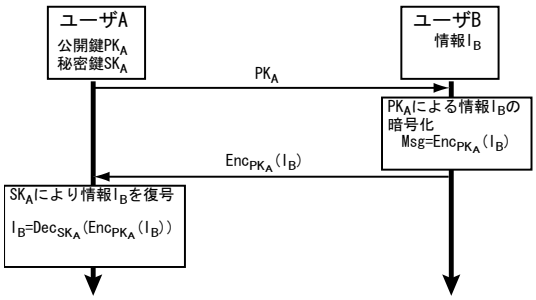

図 1: 公開鍵暗号のナイーブな利用

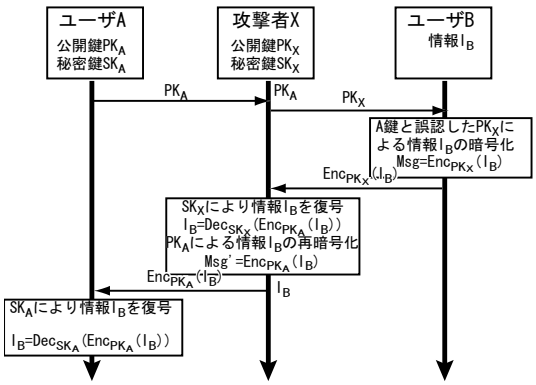

図 2: MITM 攻撃概略

ることは計算量的に困難となるよう構成されているため，暗号化用鍵は文字通り公開できる．

このような特徴を持つ公開鍵暗号を用いて安全な通信を実現しようとする場合，もっともナイーブな構成として図 1 に示すように

1. A が自身の公開鍵を B に送付
2. B は同公開鍵でデータを暗号化した上で A に送付

というものが考えられる．

懸念されるのが単純な盗聴者のみである場合には，このような構成でも A と B は十分安全に通信できる．しかし，たとえばインターネットのようにより積極的な攻撃が懸念されるような環境では，通信されるデータに対して仲介者が適宜改竄を加えた上で流通させるという MITM ({Man,MIG,Monkey}-In-The-Middle) 攻撃を受ける可能性があり，このナイーブなプロトコルは破綻を来たす (図 2)．

図 2 で攻撃者 X は A 公開鍵を送信途上で奪取した上で自らの公開鍵を A 公開鍵と偽って B に送っている．B は A のものと誤認した X 公開鍵で暗号化したメッセージを送り，X は自らの秘密鍵で当該メッセージを復号して内容を知った上で改めて A の公開鍵で暗号化したメッセージを A に送ることで全体としての辻褄を合わせる．A/B それぞれの立場に立つ限りでは送った／送られた鍵を用いた形で通信が行われているようにしか見えないため，このような攻撃者の存在に気付くことはできず，したがって安全と誤認することになる．

A/B およびその間を繋ぐ潜在的に積極的攻撃が可能な通信路以外を仮定せず，このような MITM 攻撃実行者 (=MITMer) を検出することが本論文の目的である．なお，そもそも MITM 攻撃が可能な主体にとっては，正規通信者の通信を妨げる DoS (Denial of Services) 攻撃を実行することは原理的に極めて容易であるため，本稿では MITMer による DoS 攻撃に関しては考慮の対象としない．あくまでも図 2 のような形で安全な筈の接続を介して通信されるメッセージの内容が攻撃者に筒抜けとなるような事態を回避することが目的である．

実際にこのような危険が伴う形で公開鍵の配布が行なわれる例としては，SSH サーバ [2]，SSL/TLS サーバ (無名サーバモードでの利用時)[3]，Zebedee サーバ/クライアント [4] 等，枚挙にいとまがないほどである．

2 既存対策とその問題点

ここでは現存する MITM への対策を紹介し，それらの問題点を述べる．

2.1 別経路の利用

これは MITM 攻撃が懸念される通信路とは異なる安全な経路を用いて公開鍵ないし公開鍵検証情報を配布しておくことで危険を回避しようという考え方である．

たとえば SSH の実装は，いずれもクライアント側の特定ファイル等にサーバ公開鍵を登録しておき，接続要求時にサーバから送られる公開鍵と当該登録済公開鍵とを比較することでサー

バを認証する．あらかじめ公開鍵が登録されていないサーバに初めて接続する際には，サーバから送付された公開鍵の “指紋”[1]が表示されるため，クライアントプログラム利用者は別途安全な経路で入手した当該サーバ公開鍵の指紋値を表示値と比較することで確実な接続を構築することができる[2]．

さらに，公開鍵を広範囲に安全に配布する手法として，中途での改竄が困難である放送電波を用いるというものも提案されている [5]．

この対策の最大の問題点は，事前に公開鍵もしくは公開鍵検証情報を別経路で入手しなければならないというところにあるだろう．一般にそのような別経路というのは主たる通信路と比較して使い難いことが多いし，思い立ったとき直ちに使えるというものでもない．また，公開鍵暗号利用プロトコルを安全に用いるためにはある程度の頻度で鍵ペアを更新することになるが，この方式の場合その度に登録公開鍵情報を更新しなければならないというのも厄介である．

2.2 所持者 ↔ 公開鍵の結び付き保証

もう一つの典型的手法は，所持者と公開鍵との結び付きを検証できる形で保証することである．ウェブでの http over SSL/TLS プロトコルにおけるサーバ認証はこの方法を用いている．

SSL では，サーバ公開鍵は認証機関 (Certification Authority, CA) によって発行される X.509 公開鍵証明書 [6] という形でクライアントに送付される．公開鍵証明書とは公開鍵および所持者情報等に対して CA がデジタル署名を施したものであり，一般的なブラウザに同梱されている CA 公開鍵により当該証明書の署名を検証することで，クライアントは確実にサーバ公開鍵を入手できる．

また，PGP[7] では “*Web of Trust*” と呼ばれる公開鍵流通手法が用いられる．これは X.509 と同様，デジタル署名を用いて安全に公開鍵と所持者との結び付きを保証する仕組だが，X.509 の場合とは異なりその保証は CA が独占的に行なうのではなく，ユーザが他のユーザに対して行なうようになっている．

さらに，証明書とデジタル署名を用いて公開鍵を確実に配布するための手法は，*Station-to-Station Protocol* として一般化されている．

このような仕組は一般に PKI (Public Key Infrastracture) と呼ばれるが[3]，その欠点としては，まず，大がかりな基盤 (X.509 型の場合は認証機関の構築と運営，PGP 型の場合は草の根的に大規模なユーザベース) が必要となるという点が挙げられる．そして，この方式の最大の欠点はそれが根本的な解決とはなり得ないという点にある．PKI とは結局のところ，ある公開鍵の真性性を他の公開鍵[4]の真性性に依拠して確かめているに過ぎず，それら検証用の鍵を安全に入手する普遍的な方法は存在しない．

3 提案方式

ここでは，MITM 攻撃検出対策として提案する方式を説明する．

3.1 前提

まず，本方式が前提とする環境は以下の通りである．

1. 攻撃者は通信者のいずれかに完全に成りすますことはできない．
2. 通信者は両者のみが秘密に保持する情報 (= パスワード/パスフレーズ) を共有している．
3. 共有秘密情報は暗号学的に見た場合，“弱い” 可能性がある．

1. は MITM 攻撃ではなく，通信者どちらかの環境を完全になり済ますという形の攻撃は行なえない，という条件である．

2. は計算機の遠隔利用，メールの受信，さらにはウェブによるオンラインショッピング等の

[1]公開鍵の一方向ハッシュ結果

[2]プロトコル定義上および一部の実装では後述の PKI を用いた公開鍵配布手段も提供

[3]特に X.509 による枠組を指すものとして PKI という単語が用いられることも多い

[4]X.509 型の場合は CA の，PGP 型の場合は他ユーザの

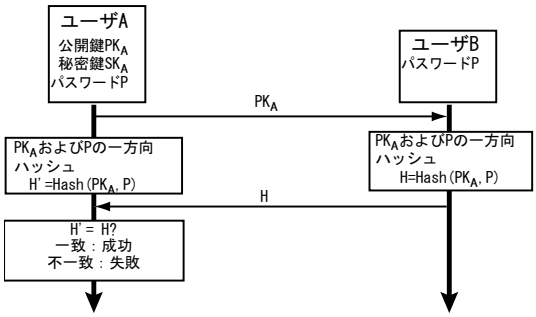

図 3: 方式 1

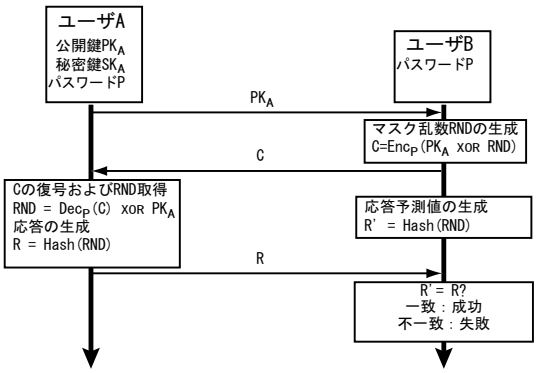

図 4: 方式 2

場面ではごく一般的に行なわれていることであり，十分普遍的な前提であろう．そして，この条件は正規の通信者間だけでの秘密の共有を意味するため，1. の攻撃をより困難なものとすることにも繋がる．

3. は，パスワードは人間が記憶するものであることから必然的に導かれる前提である．一般にパスワードの持つ複雑性は 10〜40 ビット程度となることが報告されている [9]．40 ビットの複雑性は，現状一般的な PC の計算能力を集中させれば数日程度で全数探索が行なえる程度のものであり，暗号学的観点からは弱いとみなされる．そのため，プロトコル的に全数探索が行なえないような構成を取ることが望ましい．

3.2 方式 1

このような前提のもとで MITM 攻撃を検出するための最も単純な方式は，図 3 のようなものである．この方式の主眼は，受信した公開鍵の値が送信者が送ったものと同じであることを，共有秘密情報も交えて一方向関数を適用した結果を比較することで確認する，という点にある．いわば，MAC (Message Authentication Code) による公開鍵の検証作業であり，デジタル署名により公開鍵検証を行なう前述した Station-to-Station Protocol の変形とも見なせるだろう．

図 2 を見れば明らかな通り，MITM 攻撃が実施される場合，受信者が得る公開鍵は送信者の正規の公開鍵とは別物となってしまう．そのため，本方式のように送・受信者がそれぞれ公開鍵と共有秘密情報とから MAC を計算すると，両者は異なる値となるため，その二つの値を比べることにより MITMer の有無を判断することができる．前提条件より MITMer は秘密情報を知らないため，送信される MAC 値を辻褄が合うように修正することもできない．

この方式により前項の 1. と 2. の条件は満たされるが，3. の条件はクリアできない．なぜなら，この方式では通信される公開鍵および MAC は攻撃者に筒抜けとなるため，それらの値よりオフラインで[5]秘密情報を全数探索することが可能となるからである．したがって，本方式は秘密情報として弱いパスワードが用いられた場合，容易にパスワードの値を知られてしまうこととなる[6]．

3.3 方式 2

前方式の欠点を解消するものとしては，図 4 のような方式が考えられる．

この方式では公開鍵を疑似乱数値でマスクした上で共有秘密情報で暗号化しているため，前方式とは異なり，攻撃者に対してオフラインでの計算を行なうために必要となる手がかりを与えずに済んでいる．なぜならば，比較されるのは公開鍵と同じサイズの疑似乱数値を一方向ハッシュした結果であるため，乱数値 ↔ ハッシュ値のマッピングは一意には定まらず，したがってある特定のメッセージ対を生成する可能なパスワードは無数に存在し得るからである．

また，前方式と比較してメッセージが一回余

[5] 正規の通信者に対して対話的処理を行なうことなく

[6] これは APOP，CHAP 等のチャレンジ&レスポンス型認証に共通して存在する脆弱性

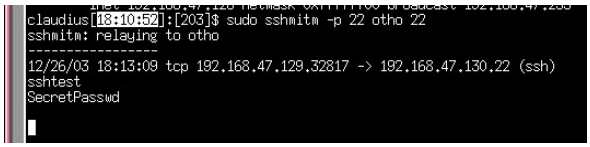

図 5: 通常 SSH への sshmitm の適用

分に必要とされるが，MITMer 不在の確認と同時に認証を実施することで，この余分なメッセージは隠蔽可能なことが，次に述べる SSH での実装の例で明らかになる．

4 SSH への試験実装

ここでは，MITM 脆弱性のあるプロトコルの代表として SSH を例に取り，方式 2 を実装した結果を報告する．前述の通り SSH は別経路で事前に入手した公開鍵[7]を用いて，セッション構築時に送信されるサーバ公開鍵の検証を行なうことが基本となっているため，あるサーバに初めて接続する際に MITM 脆弱性が存在する．また，そのような脆弱性を突いて SSH を攻撃するためのツール “*sshmitm*”[10] も公開されているため，改良策の効果を実証しやすいというのも SSH を選択した理由である．ただし，現状では同ツールは SSH1 バージョン 1 系プロトコル (以降，SSH1 と呼称) のみ対応のため，本試験実装でも SSH1 を対象とすることにした．図 5 に *ssh-mitm* を SSH プロトコルに適用した結果を示す．ユーザ名とパスワードのみを表示しているが，すべての通信内容は同様に攻撃者に対して筒抜けとなる．

また，SSH1 ではサーバ公開鍵の送信 (およびサーバ認証)，セッション鍵の交換，そしてクライアント認証が異なるフェーズとして実施されるようになっており，本試験実装ではそのクライアント認証フェーズにおけるパスワード認証と同時にサーバ公開鍵の検証を行なうこととした．このように公開鍵の送信とクライアント認証を独立したフェーズで行なうのはかなり一般的なプロトコル構成法であるため，本手法はそれら他のプロトコルにも容易に適用可能と考える．

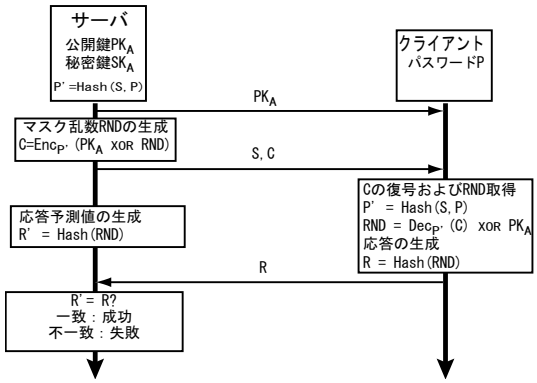

図 6: SSH1 への実装

通常の SSH サーバが実行される UNIX 系 OS では，ユーザ認証用のパスワードは撹拌要因である “*Salt*” と共に一方向ハッシュされた形でサーバ上のデータベースに格納される．そこで，パスワードではなくそのデータベース上の値を暗号化鍵としなければならないが，パスワードそのもののみを知るクライアントが当該鍵を得るためには *Salt* の値をサーバに教えてもらう必要がある．そのため，図 4 とは逆に，公開鍵の疑似乱数値によるマスクおよび暗号化は同公開鍵所持者であるサーバが実施し，*Salt* と同梱してクライアントに送信するような方式とした (図 6)．これにより，パスワードの一方向ハッシュに *Salt* を用いる UNIX 型の認証方式でもメッセージの一往復のみでクライアント認証まで実施可能となっている．もちろん，この方式ではサーバ側が不正をすればクライアントに正しく接続したと誤認させることが可能だが，3.1 章の条件 1. より完全なり済ましは不可としているように，クライアントはそのような状況に気付けることを仮定している．

前述の *sshmitm* を用いて改良版 SSH の通信に対して MITM 攻撃を行なった結果を図 7 に示す．図 5 とは異なり，改良版ではパスワード認証処理を通じてパスワードが露出することはなく，正しいパスワードでの認証が失敗するため MITMer の存在が発覚することとなり未然の防御が可能となった．

[7] もしくはその検証用情報

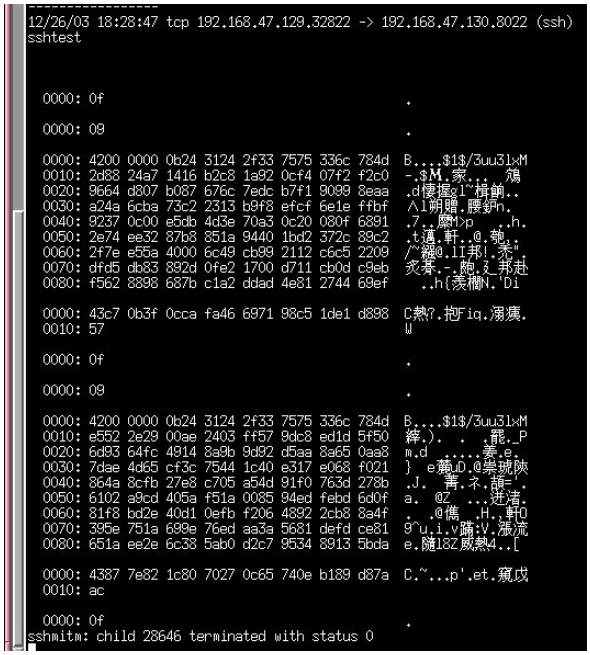

図 7: 改良版 SSH への sshmitm の適用

5 おわりに

多くの運用環境で一般的な共有パスワードを用いることで，公開鍵暗号利用プロトコルに対して MITM 耐性を与えるための一般的な手法を提案した．そして，同手法を実際に MITM 脆弱性が存在する SSH プロトコルに適用することでその現実性を示した．

MITM 対策という文脈ではないが，同様に公開鍵とパスワードを併用することで辞書攻撃に対して安全なプロトコルを構成するための手法として EKE (Encrypted Key Exchange)[11] が存在する．EKE ではパスワードを用いて暗号化した一時的公開鍵が配布されるが，このような方式では公開鍵の配布というプロトコルの根幹部に対して修正を加えなければならないことになる．それに対して本方式の場合は公開鍵配布部分は従来通りで良く，その後に行われる認証部分に後付できるため，既存プロトコルに組み込むことははるかに容易であろう．

半面，EKE は公開鍵そのものを一時的に生成するため，疑似乱数値でマスクしてあるとはいえ公開鍵そのものは既知情報となる本方式が同等の安全性を保てるかは定かではない．本稿では危険となり得る辞書攻撃に関して定性的にしか述べていないが，今後は可能ならばその部分に対して安全性の証明を付与することにより，より安心できる方式を実現したいと思う．

参考文献

[1] W. Diffie and M. Hellman: *New directions in cryptography*, IEEE Transactions on Information Theory, IT-22(6), Nov. 1976.

[2] IETF Secure Shell page: http://www.ietf.org/html.charters/secsh-charter.html

[3] IETF Transport Layer Security page: http://www.ietf.org/html.charters/tls-charter.html

[4] Zebedee web site: http://www.winton.org.uk/zebedee/

[5] 星野 春男，湯山 一郎: 公開鍵配布方法およびこの方法に用いる公開鍵送信装置ならびに公開鍵受信装置, 特開 2002—152189.

[6] ITU-T Recommendation X.509 (1997 E): Information Technology — Open Systems Interconnection — The Direcotry: Authentication Framework, Jun. 1997.

[7] IETF OpenPGP page: http://www.ietf.org/html.charters/openpgp-charter.html

[8] W. Diffie, P. C. van Oorschot and M. J. Wiener: *Authentication and Authenticated Key Exchanges*, Designs, Codes and Cryptography, vol. 2, 1992.

[9] R. E. Smith 著, 稲村 雄監訳: 認証技術—パスワードから公開鍵まで, オーム社, Apr. 2003.

[10] Dsniff web site: http://www.monkey.org/ dugsong/dsniff/

[11] S. Bellovin, M. Merritt: *Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks*, In Proc. of Winter'91 USENIX, 1991.