Vol.2015-CSEC-69 No.22
Vol.2015-IOT-29 No.22
2015/5/22

Man In The Browser 攻撃対策を実現する 人間・サーバ間のセキュア通信プロトコル

土屋貴史[†1]　藤田真浩[†2]　高橋健太[†3]　加藤岳久[†4]　間形文彦[†5]
勅使河原可海[†6]　佐々木良一[†6]　西垣正勝[†1]

Man In The Browser 攻撃（MITB 攻撃）は，ブラウザのマルウェア感染が原因であるため，SSL のような機械（ブラウザ）と機械（銀行サーバ）間のセキュア通信では対策不可能である．そこで本稿では，取引を行う「人間（ユーザ）」という計算資源に着目し，通信路におけるクライアント側の処理を人間が担うことによって，人間（ユーザ）とコンピュータ（銀行サーバ）間のセキュア通信を実現して取引改ざん型 MITB 攻撃への対策を試みる．その第一歩として，人間・銀行サーバ間でセキュア通信を実現するチャレンジ&レスポンス方式のプロトコルを提案する．提案プロトコルに対して安全性検討を行い，銀行サーバからユーザへチャレンジを「(ブラウザに潜む）マルウェアが盗聴できない通信チャネル」を通じて送信できるという前定のもとで人間・銀行サーバ間でセキュア通信を実現可能であることを検証した．同時に，「マルウェアが盗聴できない通信チャネル」は CAPTCHA を応用して実装可能であることを示すことで，提案プロトコルの妥当性を確認した．

Secure Communications Protocol Between Humans and a Bank Server to Prevent Man In The Browser Attack

TAKASHI TSUCHIYA[†1]　MASAHIRO FUJITA[†2]　KENTA TAKAHASHI[†3]
TAKAHISA KATO[†4]　FUMIHIKO MAGATA[†5]　YOSHIMI TESHIGAWARA[†6]
RYOICHI SASAKI[†6]　MASAKATSU NISHIGAKI[†1]

Man In The Browser Attack (MITB attack) is caused by malware that infects a web browser, hence conventional secure communication channels between a machine (web browser) and a machine (bank sever) such as SSL cannot prevent the attack. In this paper, we propose new approach to prevent MITB attack, which is constructing secure communication channels between a machine (web browser) and a human (end user). Our approach uses the user as a computational resource and he/she has to process an end side of the channel. Developing a challenge and response protocol which achieves the proposed channel, we conduct safety evaluation of the protocol. Its result shows that the protocol works safety under the assumption that the bank server sends a “challenge which malware in the browser cannot tap” to the user. Sending the challenge is feasible by applying CAPTCHA technology.

1. はじめに

近年，インターネットバンキングにおける不正送金の被害が急増している[1]．不正送金には種々の攻撃が存在するが，その中でも特に Man In The Browser 攻撃（MITB 攻撃）が注目を集めている．MITB 攻撃は，PC に感染したマルウェアがブラウザの操作を乗っ取ることで，認証情報の盗取や不正送金を行う攻撃である．

現在，多くのインターネットバンキングは，ブラウザと銀行サーバ間でエンド・エンドのセキュア通信（TLS，SSL 通信）[a]を行うことで，不正送金を対策している[2][3]．しかし，MITB 攻撃は「マルウェアが PC（ブラウザ）の操作を乗っ取る」攻撃であるため PC（ブラウザ）・銀行サーバ間のセキュア通信では対策できない．したがって，現状の対策に加えて，MITB 攻撃対策をインターネットバンキングへ導入することが急務となっている．

そこで，人間（ユーザ）とコンピュータ（銀行サーバ）の間に直にセキュア通信チャネルを構築するというアイデアに基づく MITB 攻撃対策を模索する．ユーザ・銀行サーバ間でのセキュア通信を実現することにより，取引内容改ざん型 MITB 攻撃[4]に対する耐性を持った取引が可能となる．

コンピュータ（ブラウザや銀行サーバ）は高い計算機能力をもつため，ブラウザと銀行サーバ間でセキュア通信チャネルを確保する仕組みは容易に実現可能である．暗号技術の利用がその典型的であり，暗号通信チャネルを利用してブラウザ・銀行サーバ間のセキュア通信を実装した報告は，既に多くなされている[2][3]．これらの報告では，高い計算能力（および記憶能力）を持つ者どうしがその計算能力をいかんなく発揮し，「計算量的安全性」に基づいて安全

†1 静岡大学大学院総合科学技術研究科
Graduate School of Integrated Science and Technology, Shizuoka University
†2 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University
†3 株式会社日立製作所 研究開発グループ セキュリティ研究部
Hitachi, Ltd., R&D Group, Security Research Dept.
†4 情報処理推進機構
Information-technology Promotion Agency
†5 NTT セキュアプラットフォーム研究所
NTT Secure Platform Laboratories
†6 東京電機大学未来科学部
School of Science and Technology for Future Life, Tokyo Denki University

a 本稿において「セキュア通信」は安全な取引を実現する通信のことをいう．具体的には，「攻撃者（ユーザ・銀行サーバ以外の第三者）　が，送金情報を改ざんすることによって，ユーザ・銀行のいずれかに金銭的な被害を与える」ことを防ぐ通信のことをいう．

Vol.2015-CSEC-69 No.22
Vol.2015-IOT-29 No.22
2015/5/22

性を確保している．一方で，ユーザ・銀行サーバ間でセキュア通信を実現するためには，人間を計算資源としてみなし，クライアント側の処理を人間が担う必要がある．しかし，人間（ユーザ）は低い計算能力（および記憶能力）しか有しないため，前述のような「(計算量的安全性に基づいた）暗号化技術を利用したセキュア通信」を利用することはできない．「低い計算能力を有するユーザ」と「高い計算能力を有するコンピュータ」間でセキュア通信を実現する手法が必要となる．本稿ではその一手法として，「(ブラウザに潜む）マルウェアが盗聴できない通信チャネル」を利用したチャレンジ&レスポンス方式のセキュア通信プロトコルを提案する．

本論文の構成は以下のとおりである．2 章で MITB 攻撃について述べた後、3 章で提案方式の説明，4 章で提案方式の考察を行う．5 章で関連研究・技術を紹介し，最後に 6 章で本論文をまとめ，今後の課題を述べる．

2. インターネットバンキングと MITB 攻撃

2.1 インターネットバンキングの送金プロトコル

本稿で想定するインターネットバンキングにおける送金プロトコルについて述べる．ここでは簡単のため、仕組みを単純化して説明している．

2.1.1 エンティティ

インターネットバンキングにおける送金プロトコルの構成要素（エンティティ）は以下の通りである．

銀行サーバ：インターネットバンキングサービスを提供する金融機関のサーバである．本稿では，銀行サーバは安全性が確保されているものとする（たとえば，サーバ内のデータが漏洩されたり，処理が改ざんされたりすることはない)．銀行サーバはコンピュータであるため，高い計算機能力（および記憶能力）を有する．

ユーザ：インターネットバンキングサービスを利用する顧客である．送金処理を実行する際には，金融機関が提供する送金プロトコルに従って PC の操作を行う．ヒューマンエラーは起こさない（想定されていない操作は行わない）ものとする．ユーザは人間であるため，低い計算機能力（および記憶能力）しか有さない．

PC：キーボード，ディスプレイを備えており，インターネットを介して銀行サーバに接続されている．PC にはブラウザがインストールされており、ユーザはブラウザを利用してインターネットバンキングの操作を行う．PC（実際には，ブラウザ）はコンピュータ（実際には，コンピュータ上のソフトウェア）であるため，高い計算機能力（および記憶能力）を有する．

2.1.2 手順

インターネットバンキングにおける一般的な送金プロトコル（図 1）の手順を述べる．

Step 1. ユーザは送金情報 X（たとえば，口座番号や送金金額）を PC へ入力する．
Step 2. PC は X を銀行サーバへ送信する．
Step 3. 銀行サーバは送金内容の確認を行うために，確認情報 Y を PC へ送信する．通常，確認情報 Y は受信した送金情報 X そのものである（Y=X)．
Step 4. PC は Y を受け取り，ユーザへ表示する
Step 5. ユーザは Y を読み，「Y が確かに自分の入力した送金情報 X と一致しているか否か」を確認したうえで，送金を確定してよいか最終判断する．
Step 6. ユーザは，送金を確定してよければ，送金確定（TRUE）を PC へ入力する．そうでなければ送金中止（FALSE）を PC へ入力する．
Step 7. PC はユーザが入力した TRUE または FALSE を銀行サーバへ送信する．
Step 8. 銀行サーバは TRUE を受信した場合は送金を受理する．FALSE を受信した場合は送金を中止する．
Step 9. 銀行サーバは送金結果（レシート）を PC へ送信する．
Step 10. PC は送金結果をユーザに表示する．

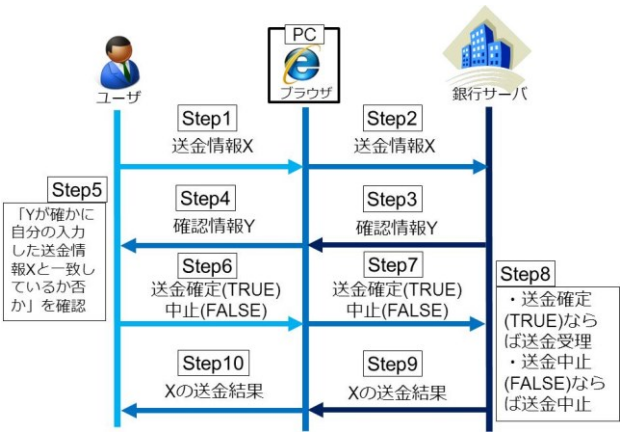

図 1 送金プロトコル

2.2 MITB 攻撃の分類

MITB 攻撃は，PC に感染したマルウェアがブラウザの操作を乗っ取ることで，認証情報の盗取や不正送金を行う攻撃である．MITB 攻撃は取引内容改ざん型と ID 盗取型に大別される[4]．それぞれの概要を以下で説明する．

2.2.1 取引内容改ざん型 MITB 攻撃

取引内容改ざん型は，ユーザが PC（ブラウザ）に入力した取引情報を、PC に潜むマルウェアが改ざんする攻撃である．これにより，ユーザが入力した情報と異なる情報によって送金取引が行われるという被害が発生する．図 1 に示

した送金プロトコルに対する取引内容改ざん型 MITB 攻撃の手順（図 2）を以下に示す．

Step 1. ユーザは送金情報 X を PC へ入力する．
Step 2. PC（ブラウザ）に潜むマルウェアは送金情報 X を X’へ改ざんし銀行サーバへ送信する．
Step 3. 銀行サーバは送金内容の確認を行うために，確認情報 Y（受信した送金情報 X’）を PC へ送信する．
Step 4. PC は Y（＝X’）を受け取る．PC に潜むマルウェアは Y を Y’（＝X）へ改ざんしたうえでユーザへ表示する．
Step 5. ユーザは Y’（＝X）を読み，「Y’が確かに自分の入力した送金情報 X と一致している」ことを確認したうえで，ユーザは送金を確定してよいかを最終判断する．Step 4 で Y（＝X’）が Y’（＝X）に改ざんされているため，ユーザは「一致している」と判断することに注意されたい．
Step 6. ユーザは，送金を確定するために TRUE を PC へ入力する．
Step 7. PC はユーザが入力した TRUE を銀行サーバへ送信する．
Step 8. 銀行サーバは TRUE を受信したため，X’に関する送金を実行する．
Step 9. 銀行サーバは X’に関する送金結果を PC へ送信する．
Step 10. PC は X’に関する送金結果を受け取る．PC に潜むマルウェアは X’を X へ改ざんしたうえでユーザへ表示する．不正な X’が正しい X に戻された形で送金結果が表示されるため，ユーザは改ざんに気付かないことに注意されたい．

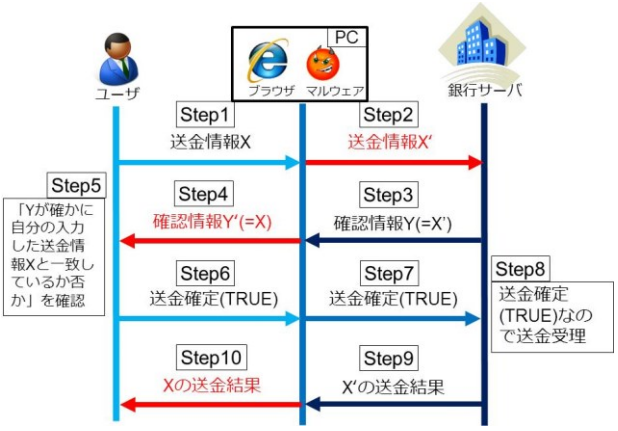

図 2 取引内容改ざん型 MITB 攻撃

2.2.2 ID 盗取型 MITB 攻撃

ID 盗取型は，ユーザのログイン時に PC に潜むマルウェアがログイン画面を改ざんし，本来は要求されない認証情報（秘密の質問や乱数表の全ての数字等）の入力を促すような偽の画面やポップアップを表示する攻撃である．これにより，ユーザが入力した認証情報は攻撃者サーバへ送信され，以降のなりすましに利用される．詳細な手順例を以下に示す．

Step 1. ユーザは、自身の PC（ブラウザ）からインターネットバンキングのサイトにアクセスする．
Step 2. PC（ブラウザ）に潜むマルウェアは、ユーザが有する秘密情報（たとえば，乱数表や秘密の質問）の入力を促すポップアップを表示するようにログイン画面を改ざんし，ユーザへ表示する．
Step 3. ユーザは、Step2 で表示されたポップアップが，銀行サーバによって表示された正規のポップアップであると誤認し，自身の秘密情報を入力する．
Step 4. マルウェアはユーザの入力した秘密情報を攻撃者サーバへ送信する．
Step 5. 攻撃者は、Step4 で盗取した情報を利用してインターネットバンキングのサイトに不正アクセスし、不正送金を行う．

3. 提案方式

3.1 コンセプト

2.2 節に示したとおり，MITB 攻撃は取引内容改ざん型と ID 盗取型の二種類に分類される．このうち，本稿では取引内容改ざん型への対策を模索する．なお、本稿でいう「対策」とは，取引改ざん型の攻撃を受けた場合でも，ユーザや銀行サーバに直接的な金銭被害がでないことを保証することをさす．（本稿ではあくまで第一報であり，今後，提案方式を拡張し，ID 盗取型への対策や間接的な被害に対する対策も実現していくことを予定している．）

現在，インターネットバンキングではブラウザと銀行サーバ間でエンド・エンドのセキュア通信（TLS や SSL）を行うことで，不正送金を対策している[2][3]．しかし MITB 攻撃は「マルウェアが PC（ブラウザ）の操作を乗っ取る」攻撃である．セキュア通信のエンドポイントである PC（ブラウザ）がマルウェアに乗っ取られているため，PC（ブラウザ）・銀行サーバ間のセキュア通信では対策できない．人間を計算資源としてみなし，人間（ユーザ）と機械（銀行サーバ）間で直にセキュア通信チャネルを構築しなければ，取引内容改ざん型 MITB 攻撃に対する根本的な対策とならない．

しかし，人間は低い計算能力（および記憶能力）しか有しないため，TLS や SSL 通信のような「（計算量的安全性に基づいた）暗号化技術を利用したセキュア通信」を利用することはできない．「低い計算能力を有するユーザ」と「高い計算能力を有するコンピュータ」の間でセキュア通信を実現する手法が必要となる．本節ではその一例として，「マ

ルウェア（機械）が盗聴できない通信チャネル」を利用したチャレンジ&レスポンス方式のセキュア通信プロトコルを示す.

3.2 提案プロトコル

3.2.1 要件

取引改ざん型 MITB 攻撃では, 図 2 の Step2(送金情報), Step4（確認情報), Step7（送金確定/中止), 9（送金結果）の通信において，マルウェアによる改ざんが行われる可能性がある．このうち、Step2、Step4、Step7 の改ざんが直接的な金銭被害に直結する（Step9 はサーバで送金が実行されてしまった後の通信である)。したがって，本稿では、Step2、Step4、Step7 で改ざんが行われたとしても，ユーザや銀行に直接的な金銭被害が発生しないことを保証するセキュア通信プロトコルを構築する.

3.2.2 マルウェアが盗聴できない通信チャネル

提案プロトコルでは,「マルウェア（機械）が盗聴できない通信チャネル」を利用することを前提とする．本チャネルの定義は下記のとおりである.

[定義] サーバからユーザへの通信チャネルが存在し，サーバがユーザへそのチャネルを用いてデータ α_1~α_n を一度に送信したとする. このときチャネルに流れたデータを{α_1, α_2,…,α_n}と表記した際,

(i) マルウェアは{α_1, α_2,…,α_n}から $\alpha_i(1 \leqq i \leqq n)$を求めることができない.

(ii) マルウェアは $\alpha_i(1 \leqq i \leqq n)$を知っていたとしても，{$\alpha_1$, α_2,…,α_n}のどの部分が α_i を表しているかはわからない.

(iii) ユーザは{α_1, α_2,…,α_n}から α_1~α_n を求めることができる.

という条件を満たす時，そのチャネルを「マルウェア（機械）が盗聴できない通信チャネル」と呼ぶ.

ここで、上記の定義はマルウェアの読取り能力に関する制約を意味しており、マルウェアがデータ α_1~α_n の値を知っている場合には、マルウェアもユーザ（人間）に対して{α_1, α_2,…,α_n}を送信すること自体は可能であることに注意されたい。

3.4 節で詳述するが、このようなチャネルを実現する有力な手段が CAPTCHA の利用である。そこで、本稿では「マルウェア（機械）が盗聴できない通信チャネル」を「CAPTCHA チャネル」と呼称することとする。

3.2.3 手順

提案プロトコルを以下に示すとともに図 3 に図示する. Step3 および Step4 で{X,R}と表記されている箇所は CAPTCHA チャネルを用いて X と R が送信されていることを意味する.

Step 1. ユーザは送金情報 X を PC へ入力する.

Step 2. PC は X を銀行サーバへ送信する.

Step 3. 銀行サーバは取引内容の確認を行うために，確認情報 Y={X,R}を PC へ送信する．ここで、R はサーバが生成した乱数である.

Step 4. PC は Y（＝{X,R}）をユーザへ表示する.

Step 5. ユーザは、Y から X と R を読み取る。ここで、Step1 でユーザが入力した X および Step3 でサーバが生成した R と、Step5 でユーザが読み取った X および R を区別するため，ユーザが読み取った X、R をそれぞれ X_u、R_u と表記する. その後，ユーザは「X_u が確かに自身の入力した情報 X と一致している」ことを確認したうえで，送金を確定してよいかを最終判断する.

Step 6. ユーザは，送金を確定してよければ，R_u を PC へ入力する. そうでなければ 0 を PC へ入力する.

Step 7. PC は、ユーザが入力した Ru または 0 を銀行サーバへ送信する.

Step 8. 銀行サーバは、受信した値が R と一致した場合，送金確定と判断して送金を受理する. 0 を受信した場合，送金中止と判断して送金を中止する. それ以外を受信した場合，異常検知と判断して送金を中止する.

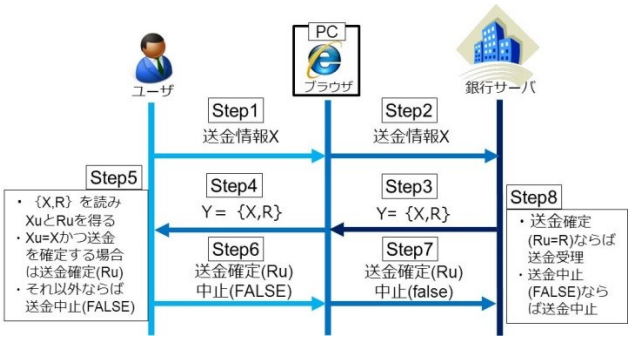

図 3 提案プロトコル

3.3 安全性検証

3.3.1 マルウェアの攻撃手法

マルウェアは Step 2、Step4、Step7 において改ざんを行うことが想定される．マルウェアが試みるであろう改ざんの全パターンを表 1 に網羅する．表 1 では，マルウェアが Step N において任意の改ざんをおこなった後に，情報が変化した状態を X_N'と R_N'と表記している.「No.」は、列挙された改ざんを呼び分けるための単なるインデックスである。

3.2.2 節の CAPTCHA チャネルの定義より，表 1 の No.4-2,4-4,7-3,7-6 に示した改ざんをマルウェアが行うことは不可能であることに注意されたい．たとえば，No.4-4 においてマルウェアが{X_2',R}から{X,R}へと改ざんするためには，①{X_2',R}を加工して{X,R}に変更するか，②自分自身で{X,R}をゼロから生成する必要がある．しかし，①に対しては、CAPTCHA チャネルの定義(ii)より、マルウェア

は{X_2',R}のどの部分が X_2'を表しているかわからず，X_2'をXへ書き換えることができない. ここで, マルウェアは、X_2'の値を知っているにも関わらず，{X_2',R}のどの部分がX_2'を表しているかは分からないことに注意されたい. 同様に，②に対しては、CAPTCHA チャネルの定義(i)より、マルウェアは{X_2',R}から R を求めることができず、{X,R}を生成することができない．また、たとえば，No.7-6 においてマルウェアが 0 から R_u へと改ざんするためには，R_u の値を知る必要がある．しかし，R_u は CAPTCHA チャネルによってサーバから送らてきている情報であるため、マルウェアは R_u の値を読み取ることができず、0 を R_u へ書き換えることができない.

表 1　Step2、Step4、Step7 で想定される改ざん

	No.	入力 (改ざん前)	出力 (改ざん後)	可否
Step2	2	X	X_2'	可能
Step4	4-1	$\{X, R\}$	$\{X, R_4'\}$	可能
	4-2		$\{X_4', R\}$	不可能
	4-3		$\{X_4', R_4'\}$	可能
	4-4	$\{X_2', R\}$	$\{X, R\}$	不可能
	4-5		$\{X, R_4'\}$	可能
	4-6		$\{X_4', R_4''\}$	可能
Step7	7-1	R_u	R_{7u}'	可能
	7-2		0	可能
	7-3	R_{4u}'	R_u	不可能
	7-4		R_{7u}'	可能
	7-5		0	可能
	7-6	0	R_u	不可能
	7-7		R_{7u}'	可能

3.3.2 安全性検証の結果

表 1 に示したマルウェアの攻撃パターンの各々に対し，Step 2、Step4、Step7 における改ざんの発生パターンの組み合わせを網羅し（すなわち、マルウェアが Step 2 のみを改ざんした場合、Step 2 と Step4 を改竄した場合、・・・のすべてのケースを想定し), すべての場合においてセキュア通信が実現されている（ユーザおよび銀行に直接的な金銭被害が発生しない）ことを確認した結果を表 2 にまとめた. 提案プロトコルにおいては、Step7 のデータが R である場合のみ送金が実行される．表 2 を見ると Step7 のデータが R となっている改ざんパターンは存在せず, Step8 で取引中止か異常検知のいずれかに至る．本結果より，提案方式がセキュア通信を実現できていることが示された.

3.4 CAPTCHA を利用した「マルウェアが盗聴できないチャネル」の構成例

前節に示したプロトコルは，サーバからユーザへ「マルウェア（機械）が盗聴できないチャネル」でチャレンジ（{X,R}）が送信できるという前提の下で設計されている．「マルウェアが盗聴できないチャネル」を実現する手法は，種々の手法があることが期待されるが，本節ではその一手法として CAPTCHA を応用した手法を説明する.

CAPTCHA は機械と人を判別するチューリングテストである[5]．ユーザへ「人間には正解容易であるが，機械には正解困難な問題」を出題することで，正解できたユーザを人間として判定する．以下，解答が $\beta_1, \beta_2, \cdots, \beta_n$ である CAPTCHA の問題を $C(\beta_1, \beta_2, \cdots, \beta_n)$と定義する．（C は，関数とは全く異なる概念であることに注意されたい.）

3.4.1 構成手法

ユーザは，n 桁の送金情報 X を X_0~X_{n-1} という n 個の数値に分割し，n 回にわたって図 3 の操作を繰り返すことで送金情報を送信するものとする. ユーザから X_i が届いたとき，銀行サーバで $C(X_i,R_i)$という CAPTCHA を生成し，ユーザへ送信する．ユーザは $C(X_i\ R_i)$を読み，X_i と R_i を抽出する．ユーザはサーバへ R_i を返却する.

ただし, ここで用いる C(X,R)は 3.2.2 節で満たした(i)~(iii)の定義を満たすような CAPTCHA を用いなければならない. 具体的には，下記のとおりである.

定義(i)を満たすために，C(X,R)は機械が解読不可能な CAPTCHA でなければならない（解読可能であった場合，機械は C(X,R)を解くことで X,R が盗聴可能となってしまう)． CAPTCHA の中には，機械によって解読可能である報告がなされているもの多く存在しており[6][7], それらは C(X,R)として用いることができない.

定義(ii)を満たすためには, マルウェアが C(X,R)のどの部分が X を表しているか認識できない CAPTCHA でなければならない．この定義を満たさない具体例として，図 4 に示す Cognometric 形式[8]の C(X,R)がある．マルウェアは，ユーザが入力した X を知っており（図 3 の Step 1)，どの画像が X を表しているかを認識することができる．実際，図 4 の CAPTCHA はマルウェアによって改ざんが可能である[b].

定義(iii)は, CAPTCHA は「人間には正解容易である問題」であるため，C(X,R)はこの定義を満たしている.

b　マルウェアが Step 2 で X'を銀行サーバへ送ると，銀行サーバから C(X',R)が送られてくる．C(X',R)は X'に直立した画像が含まれ，R に倒立した画像が含まれた CAPTCHA である．マルウェアは，X'の画像を「直立も倒立もしていない画像のペア」にし，X の画像を「直立した画像と倒立した画像のペア」にすることで C(X,R)を生成可能であり（改ざんに成功)，これをユーザに送ることで不正取引に成功する．(ただし，X=R のときには失敗する)

Vol.2015-CSEC-69 No.22
Vol.2015-IOT-29 No.22
2015/5/22

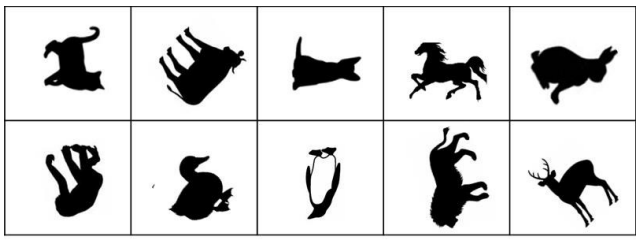

図 4　Cognometric 形式 C(X,R)の例．左上から右に向かって順に 0,1,2,3,4，左下から右に向かって 5,6,7,8,9 に対応する．そのうち，直立している画像が含まれている画像が Xu，逆立ちしている画像が含まれている画像が R を表している（今回の場合，Xu=3，R=7）．

表 2　改ざんが行われた際のプロトコル内のデータ

	No.	Step1	Step2	Step3	Step4	Step5	Step6	Step7	Step8
Step2	2	X	X_2'	$\{X_2', R\}$	$\{X_2', R\}$	$X_{2_u}' \neq X$	0	0	取引中止
Step2,4	2,4-5	X	X_2'	$\{X_2', R\}$	$\{X, R_4'\}$	$X_u = X$	R_{4_u}'	R_{4_u}'	異常検知
	2,4-6	X	X_2'	$\{X_2', R\}$	$\{X_4', R_4'\}$	$X_{4_u}' \neq X$	0	0	取引中止
Step2,7	2,7-7	X	X_2'	$\{X_2', R\}$	$\{X_2', R\}$	$X_{2_u}' \neq X$	0	R_{7_u}'	異常検知
Step2,4,7	2,4-5,7-4	X	X_2'	$\{X_2', R\}$	$\{X, R_4'\}$	$X_u = X$	R_{4_u}'	R_{7_u}'	異常検知
	2,4-5,7-5	X	X_2'	$\{X_2', R\}$	$\{X, R_4'\}$	$X_u = X$	R_{4_u}'	**0**	取引中止
	2,4-6,7-7	X	X_2'	$\{X_2', R\}$	$\{X_4', R_4'\}$	$X_{4_u}' \neq X$	0	R_{7_u}'	異常検知
Step4	4-1	X	X	{X,R}	$\{X, R_4'\}$	$X_u = X$	R_{4_u}'	R_{4_u}'	異常検知
	4-3	X	X	{X,R}	$\{X_4', R_4'\}$	$X_{4_u}' \neq X$	0	0	取引中止
Step4,7	4-1,7-4	X	X	{X,R}	$\{X, R_4'\}$	$X_u = X$	R_{4_u}'	R_{7_u}'	異常検知
	4-1,7-5	X	X	{X,R}	$\{X, R_4'\}$	$X_u = X$	R_{4_u}'	0	取引中止
	4-3，7-7	X	X	{X,R}	$\{X_4', R_4'\}$	$X_{4_u}' \neq X$	0	R_{7_u}'	異常検知
Step7	7-1	X	X	{X,R}	$\{X, R\}$	$X_u = X$	R_u	R_{7_u}'	異常検知
	7-2	X	X	{X,R}	$\{X, R\}$	$X_u = X$	R_u	0	取引中止

3.4.2 CAPTCHA の具体例

提案プロトコルに利用する定義(i) (ii) (iii)を満たす CAPTCHA の例を図 5 に示す． CAPTCHA（図 5）は直立している画像と逆立ちしている画像，それ以外の画像からなり，直立している画像の数が Xu，逆立ちしている画像の数が Ru となる．

本 CAPTCHA が定義(i)(ii)(iii)を満たすことについて示す．定義(i)は，機械が C(X,R)から Xu，Ru を得ることができない場合，達成できているといえる．本 CAPTCHA は画像が直立している，もしくは逆立ちしているか否かを判別する能力が必要である．機械はこの能力を有していない．したがって，定義(i)は満たされている．次に定義(ii)だが，X や R に紐づく画像を判別するには画像が直立している，もしくは逆立ちしているか否かを判別する能力が必要であり，機械はこの能力を有していない点から定義(ii)を満たしているといえる．定義(iii)については，人間ならば画像が直立している，もしくは逆立ちしているか否かを判別することが可能であるため C(X,R)から Xu，Ru を得ることができるため定義(iii)は達成されている．

ここで紹介した CAPTCHA はあくまで一例であり，今後提案プロトコルでの使用に適した CAPTCHA の検討が急務となる．

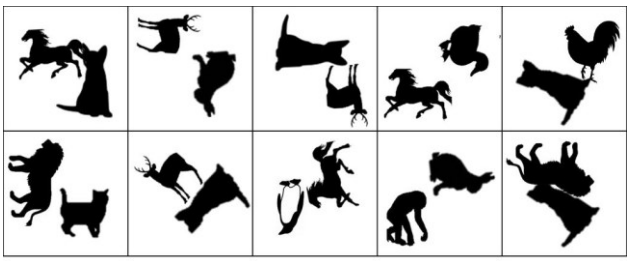

図 5　使用可能な CAPTCHA 例．直立している画像の数を Xu，逆立ちしている画像の数を Ru とする(今回の場合，Xu=6，Ru=4)

4. 考察

4.1 ランダムに改ざんする攻撃に対する安全性

提案プロトコルでは，Step 8 においてユーザから返却された R_u が銀行サーバの設定した R と一致するか否かで正

当な取引か否かを判定している．したがって，マルウェアが Step 7 において R_u をランダムに決定して改ざんした際（R_7'とする），R_7'=R となった場合は，その取引は正当な取引として処理される．すなわち，マルウェアは「1÷R が取り得る値の総数」の確率で改ざんに成功することとなる．

本稿 3.4.2 節に示した C(X,R)を用いたセキュア通信では，n 桁の送金情報 X を X_0~X_{n-1} という n 個の数値に分割し，n 回にわたって通信を繰り返すことを想定している．また，各桁に対して，R の取り得る値は 0~9 の 10 通りである．よって，1 桁に対するランダムな改ざんの成功確率は 1/10 である．

任意の口座へ不正送金を成功させるには n 桁全てで成功する必要があるため, 各桁に対する R をランダムに選択し，任意の口座へ不正送金が成功する確率は(1/10)^n となる．一般的な口座番号の長さは 7 桁であるため，口座番号を任意の値に書き換える際の成功確率は(1/10)^7 となり，非常に小さい．不正送金の目的の多くが，攻撃者の利益を得ること（ユーザが入力した口座番号を自身の口座番号へ書き換えること）に鑑みると，提案手法は十分に高い有効性をもつといえる．

しかし，1 桁だけ送金情報を改ざんし，ユーザの意図した口座以外への不正送金を行う攻撃や送金金額を改ざんする攻撃も考えられる．このような 1 桁だけランダムに変更するような攻撃の成功確率は 1/10 であり比較的大きな値である. この問題に対する対策としては, 図 3 における Step 3~7 の操作を各桁につき m 回行う方法がある．これによって，1 桁当たりの成功確率は(1/10)^m まで減らせるが，それに伴って利便性の低下も推測される．安全性（ランダムに変更する攻撃への耐性）と利便性（プロトコルを行うユーザ負荷）のバランスに関しては今後検討を進める必要がある．

また，3.4 節に示した手法はあくまで一例であり，提案プロトコルを実現する手法は多岐に渡るものと推測される. そのほかの手法を探ることで，より大きな範囲で R を取り得る{X,R}の実現手法が見つかる可能性もある．

4.2 ユーザビリティ

本稿 3.4.2 節に示した C(X,R)を用いたセキュア通信では，n 桁の送金情報 X を X_0~X_{n-1} という n 個の数値に分割し，n 回にわたって通信を繰り返すことを（現時点では）想定している．ユーザは提案プロトコルで送金処理を行う際に n 個の CAPTCHA を解く必要があり，従来の送金プロトコル（図 1）と比較して，ユーザに対して負荷を与える方式となっている．本方式がユーザにとってどの程度の負荷であるかは，評価実験を通して検証する必要がある．

5. 関連技術・研究

5.1 マルウェア対策

MITB 攻撃はマルウェア感染が原因であるため，専用ソフトウェア（たとえば，Phish wall プレミアム[9]）によってユーザ端末上でマルウェアを検知・削除することで対策可能である．しかし，マルウェアの亜種が日々作成されている現状に鑑みれば，その効果は限定的であるといえるだろう．実際，通常のパターンマッチング方式のマルウェア対策ソフトでは平均検知率は 50~60%であり，パターンの更新速度がマルウェアの進化速度に追いついていないという現状が報告されている[10]．提案方式はマルウェアを検知するのではなく，マルウェアに感染していたとしても不正な取引が行われない方式である．すなわち，マルウェアの検出率は問題とならない．

5.2 取引内容防護

5.2.1 トランザクション署名

トランザクション署名は取引時に「PC とは独立したセキュアなハードウェア」（以後，トークンと呼ぶ）を用いて MITB 攻撃を防止する対策である[11][12]．ユーザは送金情報送信時に，トークンを用いて送金情報から確認コードを生成し，送金情報と確認コードのペアを銀行サーバへ送信する．それらを受け取ったサーバは，受け取った送金情報と確認コードの整合性を検証することで送金情報の完全性を確認する．

現在までに，銀行から配布された機器をトークンとして用いる方式[11]が提案されている．しかし本方式では，配布された機器を常に携帯する必要があるため，機器紛失や送金機会損失が発生する可能性がある．加えて，すべてのユーザへ機器を配布するには膨大なコストが必要である．

トークンとしてスマートフォンを用いる方式[12]であればこれらの問題は生じない．しかし， PC のマルウェア感染被害が頻発している現状に鑑みると，PC 同様，外部通信を自由に行えるスマートフォンが「セキュアな（マルウェアに感染していない）ハードウェア」であると保証することは困難であろう．仮に現時点でスマートフォンへのマルウェアの侵入経路をすべて対策したとしても，マルウェアの侵入手法は今後より一層高度化することが予想される．したがって，5.1 節に示したマルウェア対策と同じ問題を抱えることとなる．

提案方式は，図 3 に示したとおり，通常の送金プロトコル（図 1）に含まれる機器のみで実現可能である．したがって，トランザクション署名のように他機器を導入しておらず，上述のような問題は起こりえない．

5.2.2 Arcot VPS

Arcot VPS[13]は CAPTCHA を利用して MITB 攻撃対策を試みた事例である（図 6）．サーバは，ユーザから送られてきた送金情報とワンタイムパスワード（OTP）を文字判読型 CAPTCHA に埋め込んで PC へ送信する．PC にはサーバから送られてきた CAPTCHA が表示され，ユーザはそこに書かれている送金情報と OTP を読み取る．ユーザは，読み取った送金情報が自身の入力した送金情報と一致している

場合，OTP を入力して取引を確定する．サーバはユーザが入力した OTP が，ユーザへ送った OTP と一致した場合に限り，その取引を受理する．各取引には有効時間が定められており，サーバはその有効時間内にユーザからレスポンスを受け取れなかった場合，その取引を破棄する．

本方式は「OTP をマルウェアが解読不可能な形式（文字判読 CAPTCHA）でユーザへ渡す」ことを試みたものである．しかし，文字判読 CAPTCHA は機械でも短時間で解読可能であることが知られているため[6]，本方式はユーザへ OTP を渡す手段として適切でない．実際，文字判読 CAPTCHA 解読技術を利用して本方式を突破する報告もされている[14].

本方式と異なり，本稿 3.4.1 節に示した手法では，マルウェアが解読不可能な形式でユーザへ OTP（本稿では R と記した）を伝達している．また，本稿の目的はユーザ・銀行サーバのセキュア通信プロトコルを一般化することが主旨であり，CAPTCHA を応用して OTP を送ることは実現手段の一手段に過ぎない．

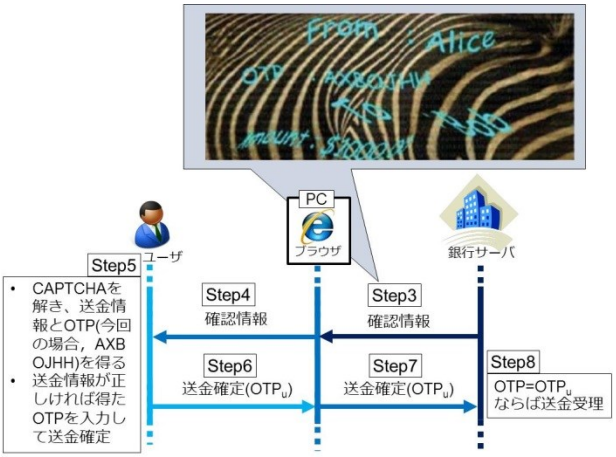

図 6 Arcot VPS

6. おわりに

6.1 まとめと今後の課題

本稿では人間（ユーザ）とコンピュータ（銀行サーバ）間のセキュア通信を実現することで MITB 攻撃への対策を試み，その第一歩として，人間・銀行サーバ間でセキュア通信を実現するチャレンジ＆レスポンス方式のプロトコルを提案した．銀行サーバからユーザへ「(ブラウザに潜む) マルウェアが盗聴できないチャネル」を通じてチャレンジを送信できるという仮定のもとで，提案プロトコルの安全性検討を行った．「(ブラウザに潜む) マルウェアが盗聴できないチャネル」が実現できるのであれば提案プロトコルにおいてユーザ・銀行サーバ間のセキュア通信が可能であることを示した．「(ブラウザに潜む) マルウェアが盗聴できないチャネル」を送る一手法として，CAPTCHA を応用した手法を示した．

今後は，提案プロトコルの実現により適した CAPTCHA の検討とプロトコルの安全性について定式化を行い，提案プロトコルのユーザビリティ調査を行う．

6.2 Secure Heterogeneous Channel

本稿では人間（ユーザ）とコンピュータ（ブラウザ）という非対称な計算機能力を有する者間におけるセキュア通信を実現した．しかし，この「非対称な計算機能力を有する者間のセキュア通信」の考え方は，MITB 攻撃対策（ブラウザ・銀行サーバ間）に限ったものでない．

IoT（Internet of Things）の分野がその一例である．IoT が実現された社会では，センサがあらゆるものに搭載されており，それらはサーバへセンサデータを送信する[15]．しかし，センサデータはプライバシー情報にあたる場合も多い．すなわち，コンピュータ（センサ）とコンピュータ（サーバ）間のセキュア通信が必須となるであろう．センサはサーバに比べて非力な計算機能力であるのは明らかであるため，この場合においても「非対称な計算機能力を有する者間（センサ・サーバ）のセキュア通信」の実現が課題となる．

このように「非対称な計算機能力を有する者間のセキュア通信」は今後様々な分野で検討が必要な課題である．筆者らはこの通信方法を「Secure Heterogeneous Channel」（セキュアな非対称チャネル）と名付けており，様々な分野で Secure Heterogeneous Channel の構築を進めていく予定である．

参考文献

1) 平成 26 年上半期のサイバー空間をめぐる脅威の情勢について，
http://www.npa.go.jp/kanbou/cybersecurity/H26_kami_jousei.pdf
2) 三菱東京 UFJ 銀行，当行のセキュリティ対策，
http://direct.bk.mufg.jp/secure/toukou.html
3) 三井住友銀行，三井住友銀行での取り組み，
http://www.smbc.co.jp/kojin/security/school/web/program.html
4) 鈴木 雅貴，中山 靖司，古原 和邦，:インターネット・バンキングに対する Man-in-the-Browser 攻撃への対策「取引認証」の安全性評価，日本銀行金融研究所 金融研究，第 32 巻,第 3 号，pp.51-76,(2013).
5) The Official CAPTCHA Site
http://www.captcha.net/
6) J.Yan, A.S.E.Ahmad: Breaking Visual CAPTCHAs with Naïve Pattern Recognition Algorithms, 2007 Computer Security Applications Conference, Vol.asss,pp.279-291,(2007).
7) Golle, P.: Machine Learning Attacks Against the ASIRRA CAPTCHA, Proc. 2008 ACM Conference on Computer and Communications Security, pp.535–542 (2008).
8) Two Factor Authentication, Graphical Passwords – Passfaces（2015 年 4 月 14 日取得） http://www.realuser.com/
9) PhishWall,
http://www.securebrain.co.jp/products/phishwall/
10) 不正送金対策 ～OCRA 仕様 OTP トークンを利用し，MITB による不正送金リスクを低減～，
http://www.ftsafe.co.jp/solutions/ocra_mitb
11) トランザクション署名：マンインザミドル攻撃(中間者攻撃) に最も有効な対策」，

http://www.vasco.co.jp/phishing/Solutions_SignaturePassword.html
12) A.SAISUDHEER, M.TECH: Smart Phone as Software Token for Generating Digital Signature Code for Signing In Online Banking Transaction, IJCES,Vol.3,No.12 (December 2013).
13) Man-in-the-Browser および Man-in-the-Middle 攻撃から オンライン顧客を保護,
http://www.ca.com/~/media/Files/whitepapers/jpProtection_from_MITM_MITB_Attacks_White_Paper201104010.pdf
14) Sang-ho Lee, Sung-ho Kim, Dea-hun Nyang, Kyung-hee Lee, : The Vulnerability Analysis of CA Arcot VPS, JKIISC, Vol.23, No.5,pp825-830,(2013)(in Korean).
15) 「Internet of Things による新ビジネスの可能性」～モノのインターネットは、企業に何をもたらすのか～
https://www.nri.com/jp/event/mediaforum/2014/pdf/forum211_4.pdf