

IDに基づく受信制御の自律管理方式とその効果

An autonomous receiver-side control mechanism based on ID information and its effect

阪本 裕介 †

Yusuke Sakamoto

中山 雅哉 †

Masaya Nakayama

1 はじめに

現在、インターネット上で電子メールやインスタントメッセージ、IP 電話といった様々なコミュニケーションツールが利用されているが、それぞれのツールにおいてスパムの増加が問題となっている [2]。スパムを識別して排除する研究は多数存在するが、意図した相手からの通信をスパムと誤判別する問題があり、自分が送信したメッセージが相手に正しく受け取られているか不安に思うユーザが存在している [5]。このような背景において、意図した相手からの通信を正しく受け取ることがより重要になっている。

その目的を達成する方法の一つとして、メッセージの送信に使われた ID (以下、コンタクト ID) を元に正当な相手からのメッセージを識別する方法がある。コミュニケーションツールでは電子メールアドレスなどのコンタクト ID が利用されておりコンタクト ID での識別は汎用性が高いが、相手のコンタクト ID を把握する必要がある。しかし、相手に応じたコンタクト ID 及びツールの使い分けは日常的に見られ、またコンタクト ID は変更されることもある。このとき、相手のコンタクト ID 把握に困難が生じ、コンタクト ID による識別ではメッセージを正しく判別できなくなる。

本研究では、コミュニケーションツールの利用背景を考慮し、相手のコンタクト ID の把握をメッセージの到着時に行うことで、受信するか否かをユーザごとに決定可能にする方法を提案する [8]。本論文では、提案手法の実現に向けて、OpenID 認証 [6] を介して各ユーザがコンタクト ID 情報を管理、取得する機構の実装を行った。この実装の元でのコンタクト ID 情報取得に必要な時間測定の結果、各コミュニケーションツールへの提案手法の追加によって生じる遅延は実用面において問題が生じない程度であることがわかった。

2 関連研究

コンタクト ID を用いて正当な相手からのメッセージを判別する手法としては、通信履歴などから各コンタクト ID に信頼度を割り当てる手法 [1][3] や、SNS (Social Networking Service) から人間関係に基づいてコンタクト ID 情報を取得する手法 [7] などがある。

信頼度を割り当てる手法 [1][3] では、ユーザが複数のコンタクト ID を使い分けていてもそれぞれに高い信頼度が割り当てられていれば、未知のコンタクト ID を用いたメッセージでも正当性を判別することが可能である。一方、スパム送信者による作為的な信頼度割り当てを防止するために新しいコンタクト ID への高い信頼度割り当てはできなくなっている。そのため、コンタクト ID の変更時に変更後のコンタクト ID を用いたメッセージを正当なものとして識別することは難しい。

SNS からの情報取得 [7] では、メッセージの送受信の際に SNS から相手のコンタクト ID 情報を逐次取得すればコンタクト ID の変更に対応した識別が可能になる。しかしながら、この研究で用いられている SNS には相手に応じて異なるコンタクト ID を公開する機能はなく、ツールやコンタクト ID を使い分けるといった利用形態への対応ができてない。

正当なユーザからのメッセージを誤判別から守るという課題を、これらの関連研究は部分的にしか解けていないと言

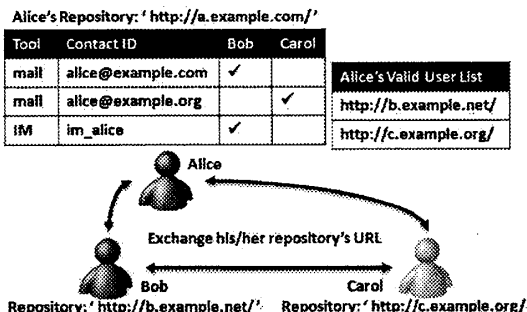


図 1: コミュニケーション ID の管理

える。

3 提案手法

本研究では、各ユーザが Web 上に自身のコンタクト ID とそれらの公開先を管理する機構 (以下、リポジトリ) を持ち、コミュニケーションツール経由でのメッセージの到着時に受信者がそこからコンタクト ID 情報を取得することで送信者を識別する手法を提案する [8]。

各ユーザは図 1 のようにリポジトリを持ち、そこで自身のコンタクト ID 群とその公開先を管理する。このリポジトリの URL を、すでにコミュニケーションツールを用いて連絡をとりあっている相手、あるいは今後そうしたい相手と互いに交換する。その後、ユーザは交換した URL を用いて正当な相手のリストを作成する。未知のコンタクト ID を用いたメッセージが到着した時に、リストに記載されている URL で表されるリポジトリ群からコンタクト ID 情報を取得し、メッセージの送信者が正当な相手かどうかを判別する。

ここで、相手に合わせたコンタクト ID の使い分けを実現するには、リポジトリがユーザごとに異なるコンタクト ID 群を見せるようにする必要がある。そのためには、リポジトリにおいてユーザの識別と認証が必要となる。提案手法では、リポジトリの URL がユーザと 1 対 1 に対応しているため、それを用いてユーザの識別を行う。ユーザの認証については、URL を認証の識別子として用いる OpenID 認証で行う。OpenID 認証を用いるために、各ユーザはリポジトリに自身の認証局 (OpenID Provider, 以下 OP) の URL を記載しておくものとする。

メッセージの送信者が自身のコンタクト ID 群や公開したい相手の変更に合わせて自身のリポジトリの情報を更新し、受信者がメッセージの到着時にそこから情報取得を行うことで、コンタクト ID の変更や使い分けが生じていても正当な相手からのメッセージを誤判別から守ることができる。

4 リポジトリの実装と情報取得の流れ

提案手法におけるリポジトリを Web サーバ上に CGI プログラムとして実装した。使用したプログラミング言語は Perl である。OP についても PHP を用いて実装した。

ユーザは自身のリポジトリにおいて自身のコンタクト ID 群の管理とその公開先を設定できる。リポジトリはユーザの設定内容とその設定の最終更新日時を記録する。この最終更

† 東京大学 The University of Tokyo

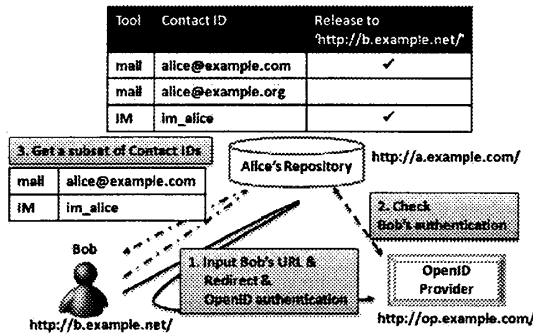


図 2: コンタクト ID 情報取得の流れ

新日時については、リポジトリにアクセスしてくるユーザ全員に認証を経ることなく公開されるようにした。

リポジトリからのコンタクト ID 情報取得の流れを図 2 に沿って説明する。

1. ユーザが自身の URL (<http://b.example.net/>) を相手のリポジトリに入力する。入力をするときリポジトリが OpenID 認証の仕様 [6] に基づいてユーザの OP の URL (<http://op.example.com/>) を取得し、ユーザは OP へ転送される。ユーザはそこで認証状態を確立させる。
2. 認証を確立させた後に、ユーザが再度リポジトリにアクセスすると、リポジトリはユーザの認証状況について OP に確認する。
3. 認証が成功していれば、リポジトリの管理者が公開設定をしたコンタクト ID 群が返され、ユーザはそれ取得する。認証の成功をリポジトリが確認できなければ、再度 1. に戻る。

以上の流れによって、自身に公開されているコンタクト ID 群を各ユーザが取得できるよう、リポジトリの実装を行った。

5 コンタクト ID 取得ツールの実装と結果

リポジトリへのアクセスは通常の Web ブラウザ経由でも可能であるが、今後各コミュニケーションツールから利用することを考慮し、リポジトリからコンタクト ID 情報を取得する機能を Perl で別途実装した。このコンタクト ID 取得ツールの動作は以下の通りである。

1. 事前に交換した URL で表されるリポジトリ群にアクセスを行い、それぞれの最終更新日時を取得する。ここで取得した最終更新日時が以前取得したものと同一の場合はここで動作を終了する。そのリポジトリへの初回アクセスの場合、または以前のアクセス時よりも最終更新日時が新しくなっている場合は、そのリポジトリからのコンタクト ID 情報取得を行うため、以下に進む。
2. 情報取得を行う際、まず OP にアクセスして認証状態を確立させる。
3. コンタクト ID 情報取得対象のリポジトリ群に対してユーザの URL を入力する。その後、4 の流れに沿ってコンタクト ID 群を取得し、ユーザに表示する。

提案手法ではメッセージの到着時に送信者を確認するため、情報取得による遅延が生じる。コンタクト ID 取得ツールを 1,000 回動作させて、時間の計測を行った。リポジトリと OP は同一ホスト A で Web サーバのバーチャルホストとして動作させ、コンタクト ID 取得ツールは別のホスト B で動作させた。各ホストについては、表 1 の通りである。両ホスト間の RTT (Round Trip Time) は 3[ms] 程度であった。

測定結果は表 2 の通りとなった。リポジトリから最終更新日時を取得しコンタクト ID の情報を取得するプロセスに進むかを判断するために 60[ms]、OP との認証状態確立に 150[ms]、リポジトリからのコンタクト ID 情報取得に 543[ms] の平均

表 1: マシンスペック

	CPU	メモリ	OS
ホスト A	Athlon 64X2 4600+	4GB	Linux 2.6.24
ホスト B	CoreDuo T2500	2GB	Linux 2.6.22

表 2: 測定結果

	平均 [ms]	標準偏差 [ms]
最終更新日時取得	60	34.7
OpenID 認証	150	24.8
コンタクト ID 取得	543	64.0

時間がかかった。すなわち、一つのリポジトリのアクセスにおいて、更新がなければ 60[ms]、更新があれば 693[ms] がかかった。RTT や情報取得対象となるリポジトリの数によってこの結果は変化するが、2 秒程度の待ち時間であれば利便性の面で許容できるという報告 [4] を考慮すると、提案手法を実際のコミュニケーションツールに追加しても問題がない程度の時間であると考えられる。

6 結論と今後の課題

本研究で提案した手法では、各ユーザがコンタクト ID の公開先を管理し、メッセージの到着時に受信者がリポジトリから情報を取得することで、送信者の識別および受信制御を行う。本手法により、コンタクト ID の違いに影響されずに、正当な相手からのメッセージを誤判別から守ることが可能になると言える。

本論文では、提案手法を実現するプロトタイプを実装し、リポジトリからの情報取得にかかる時間を計測した。認証を介してリポジトリから情報を取得する場合でも平均 693[ms] であった。これは、提案手法を実際のコミュニケーションツールに追加しても問題がない程度の時間であると考えられる。

今後は、複数リポジトリからの情報取得に必要な時間の見積りおよび短縮、リポジトリの機能拡張、各種コミュニケーションツールへの本手法の追加などを行う予定である。

参考文献

- [1] V. A. Balasubramanian, M. Ahamad, and H. Park. Call-Rank: Combating SPIT Using Call Duration, Social Networks and Global Reputation. *Conference on Email and Anti-Spam*, 2007.
- [2] V. G. Cerf. Spam, spim, and spit. *Commun. ACM*, Vol. 48, No. 4, pp. 39-43, 2005.
- [3] P. A. Chirita, J. Diederich, and W. Nejdl. MailRank: using ranking for spam detection. *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 373-380, 2005.
- [4] A. Culotta, R. Bekkerman, and A. McCallum. A study on tolerable waiting time: how long are Web users willing to wait? *Behaviour & Information Technology*, Vol. 23, pp. 153 - 163, 2004.
- [5] D. Fallows. Spam: How it is hurting email and degrading life on the Internet. *Pew Internet and American Life Project*, pp. 2008-01, 2003.
- [6] D. Recordon and D. Reed. Openid 2.0: a platform for user-centric identity management. In *DIM '06: Proceedings of the second ACM workshop on Digital identity management*, pp. 11-16, New York, NY, USA, 2006. ACM.
- [7] T. Yokoyama, S. Kashiwara, T. Okuda, Y. Kadobayashi, and S. Yamaguchi. A Generic API for Retrieving Human-Oriented Information from Social Network Services. *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on*, p. 33, 15-19 Jan. 2007.
- [8] 阪本裕介, 中山雅哉. OpenID を用いたユーザ指向型ホワイトリスト作成手法の提案. 情報処理学会研究報告, 2008-GN-67, pp. 73-78, 2008.