

EMAIL SENDER AUTHENTICATION  
DEVELOPMENT AND DEPLOYMENT

(PROJECT CHEESEPLATE)

Volume I  
Technical and Management Proposal

pobox.com  
IC Group, Inc.  
mengwong@pobox.com

v1.01 20041217

Full Proposal Control Number EB8A

## OFFICIAL TRANSMITTAL LETTER

IC Group, Inc., a New York State corporation, doing business as POBOX.COM, respectfully submits a proposal in response to solicitation BAA04-17 for Cyber Security Research and Development. It is submitted under Category 3, Technical Topic Area 7, *Technologies to Defend Against Identity Theft*, for consideration as a Type II Prototype Technology.

Solicitation Title: BAA 04-17

Topic Title: Technologies to Defend Against Identity Theft

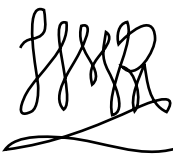
Type Title: Type II (Prototype Technologies)

Full Proposal Control Number: EB8A

Proposal Title: Email Sender Authentication

A companion proposal, *Reputation System Clearinghouse* (1RGT), is also being submitted under the same category and type. We request that these two proposals be read together. This proposal should be read first.

This proposal was authored by Meng Weng Wong, Founder and Chief Technology Officer for Special Projects. He can be contacted at [mengwong@pobox.com](mailto:mengwong@pobox.com).



Meng Weng Wong

IC Group, Inc.

1100 Vine St Ste C8

Philadelphia, PA 19107

December 15th 2004

EIN: 113236046

Central Contractor Registration: 3EKUCT

## EXECUTIVE SUMMARY

Pobox.com aims to fight phishing by adding sender authentication functionality to the Internet email system. First we will build a library to implement a useful set of recently devised anti-forgery specifications, including IP-based approaches such as SPF and crypto-based approaches such as DomainKeys. The library will also be able to query arbitrary third party reputation and accreditation services. It will be constructed as a reference implementation and documented as a standard. Then we will integrate that library into the Mail Transfer Agents (MTAs) which carry the bulk of the Internet's email. At the end of the project, it will be possible for most mail systems to simply upgrade their MTAs. After upgrading, systems can “flip a switch” and automatically recognize, block, or flag suspected spam and phishing emails. This meets the requirements of TTA 7.

In this system, receivers of email will enjoy protection from identity theft and phishing. They will have the option to easily block obvious forgeries. They will also have the option to flag mail which does not pass some form of authentication, or which does not meet minimum standards of reputation. These capabilities do not exist today in a non-proprietary, non-commercial form suitable for fast free widespread adoption. This project creates these capabilities and introduces them to the email system.

The project will last approximately eighteen months. The bulk of the money will go toward paying programmers to write software and plug it into popular MTAs both commercial and opensource. The project will be managed by the team who produced the successful SPF specification and software development effort.

Software produced by the project will be as free as is practical. Licensing will work according to generally accepted opensource practices. Participation in the antiforgery system will be voluntary and free of external costs for both senders and receivers.

Most of these benefits will not require end-user involvement. Software deployment will rely on trained system administration professionals. Portions that do affect end-users are designed to be as simple as possible and no simpler – at about the level of complexity of the web browser padlock icon.

To achieve these goals, we seek a grant between US\$150,000 and US\$750,000. The base level of funding covers the development of the core library. Higher levels of funding will allow us to upgrade more software to use that library.

“Phishing” is a class of high-tech scam that uses fraudulent e-mail to deceive consumers into visiting fake replicas of familiar Web sites and disclosing their credit card numbers, bank account information, Social Security numbers, passwords and other sensitive information.

– BAA04-17

## PERFORMANCE GOALS

The ultimate goal of this project is to change the way Internet email works. This ambition deserves explanation.

When email was invented, abuse considerations were secondary to functionality. The ability to get mail from random strangers was considered a primary virtue. Today, spam outnumbers non-spam email (ham). In the absence of filtering, messages from strangers are now, more likely than not, spam. This position may be regrettable but it reflects reality. It calls for a new paradigm of email.

While content filtering has proven effective in the past, it is unsatisfactory in a number of fundamental ways. The first generation of antispam technologies filtered out bad messages based on what they contained. The next generation of antispam technologies will filter in good messages based on who sent them.

Sender authentication is now generally considered essential to fight phishing, spam, worms, viruses, and other forms of online messaging abuse. Authentication technologies, when widely deployed and used in conjunction with reputation systems, promise to make a permanent contribution to the antispam and antiphishing effort.

Under the new paradigm, email receivers use authentication technologies to tell if the senders and authors of messages really are who they say they are (spoof detection). Then receivers use reputation technologies to check if those senders are recognized or not (stranger detection). Receivers can use these technologies together in service of policy: if a message is authenticated and recognized, then it is not spoofed and not from a stranger. Receivers can opt to treat the message positively. If the sender is not authenticated or not reputable, then the message is from a stranger. It may contain undesirable content, and receivers can opt to treat the message negatively. (The criteria used for determining “strangerness”, and how positive and negative dispositions are handled, are locally determined by individual end-users and receiver systems.) Receivers can use these technologies and apply these policies to automatically screen out unwanted messages, including phishing attempts and traditional spam.

The proposed project uses the above approach to meet the goals specified in TTA 7.

The project will implement a collection of authentication and reputation technologies and deploy them across a wide variety of software programs that make up the Internet email system.

## THE NEED FOR A COORDINATED ROLLOUT

History shows that if we leave industry to its own devices, rollout

See remarks by the FTC in the Federal Register, <http://www.ftc.gov/opa/2004/09/emailauth.htm>

Reputation systems are described in the accompanying proposal, *Reputation System Clearinghouse*.

may occur in a slow, haphazard, and “every man for himself” fashion. Such a rollout could threaten the integrity of the email system. Furthermore, some MTA offerings, particularly opensource products, may lack the resources to implement the desired improvements in a timely fashion.

If we coordinate the rollout, we can ensure that implementations meet a minimum standard of quality; we can test interoperability; and we can set a rough schedule for deployment.

After a receiver site implements sender authentication, and after a useful fraction of sending sites emit authenticated messages, it will become significantly easier for end-users to recognize phishing attempts, and for machines to automatically block them altogether.

The bulk of these technologies are intended to occur at the core of the email system and will be deployed by trained system administration professionals. Some user-visible changes to Mail User Agent (MUA) software are, however, unavoidable; they add an element to the MUA user experience roughly comparable to the padlock icon in web browsers.

## THE NEED FOR FUNDING

To end spam and phishing, the email system must evolve. Free and open standards must form the basis for this evolution because email is too important to be owned or controlled by anyone.

The single most widespread and successful such standard is SPF. In twelve months SPF has grown to cover approximately 20% of all Internet email. During that time, however, the project received only about \$3,000 in donations. This has not been enough to pay programmers, so people who work on SPF do so in their spare time.

The world seems to want a final solution to spam very badly, but it doesn't seem to want to spend much to get it. At the same time, it seems quite happy to spend billions of dollars treating the symptoms.

Money spent on treating symptoms: \$3,000,000,000.

Money given to curing the disease: \$3,000.

Yes: the most successful project to end a worldwide scourge is being run on a budget of \$3,000 by hobbyists working nights and weekends. Imagine how much better they could do if only they had a little more time and money.

With proper funding, programmers could devote more time to the project. At this point, the lack of developer resources is the single biggest obstacle to progress. All other factors are in place. It's time to stop doing this on the cheap.

Government domains are also expected to participate in sender authentication to protect themselves from receiving spoofs and from being spoofed.

Nathaniel Borenstein, Distinguished Engineer at IBM and author of the MIME specification, has counted at least thirty-one consortia chartered to fight spam. Many of them have budgets in the millions of dollars. Many of them have held expensive conferences to discuss the spam problem. None of them, to my knowledge, have allocated resources towards actually writing any antispam code.

Experts estimate direct losses from phishing at anywhere from \$150 million to \$500 million. <http://news.zdnet.co.uk/internet/security/0,39020375,39175678,00.htm>

More broadly, what does spam cost the economy? It depends who you ask. Ferris Research estimated a loss of \$11.4 billion in 2002. Other experts say that number is bogus. But it's obvious that enterprises and ISPs spend millions to handle spam volume – millions that could be better spent elsewhere, or not spent at all.

## DETAILED TECHNICAL APPROACH

There is general agreement in the industry that a permanent solution to spam and phishing will require sender authentication in combination with reputation systems and accreditation services.

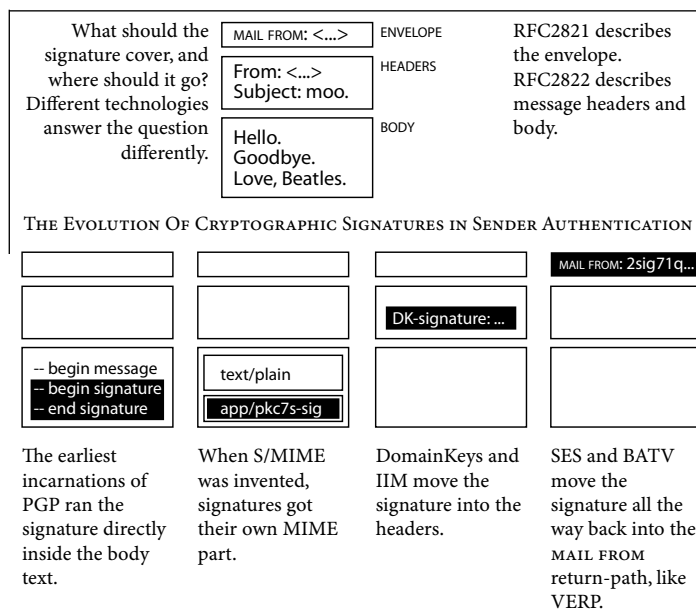
There is further agreement that there are two major schools of sender authentication: IP-based and crypto-based. The approaches are represented by a number of specifications. Each of those specifications has strengths and weaknesses. For example, IP-based systems tend to fail in cases of verbatim forwarding. Crypto-based systems tend to fail for traditional mailing lists. Furthermore, different specifications focus on different identities in the mail system. There are scenarios in which one identity may return a negative result and another identity a positive one. An ISP may use one scheme to assert that a certain network range is occupied by broadband nodes which, to the best of its knowledge, do not send mail directly to the Internet. However, one of those nodes may be operated as a Unix server by a hobbyist; that hobbyist may use a different authentication scheme to establish accountability for messages sent. In that case, a receiver may wish to execute an override.

The world is a big place. Email has many users. The use cases are many, their interactions complex. No one authentication scheme can satisfy all the requirements. We measure temperatures in Fahrenheit, Celsius, and Kelvin. We drive on both the left and the right side of the road. We can use multiple authentication schemes to help handle the complexity that exists on the Internet.

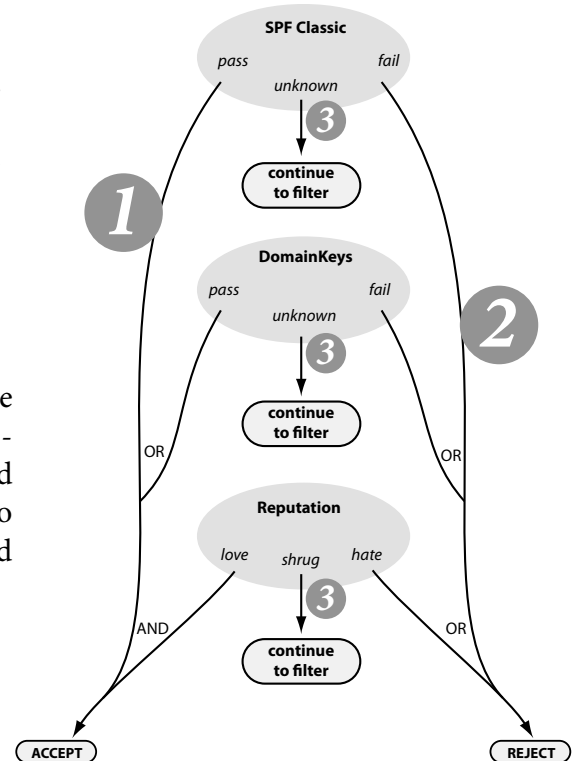
An email transaction involves many distinct identities. Some are closer to the notion of “sender”. Some are closer to the notion of “author”. Different schemes focus on different identities.

- The TCP/IP transport model authenticates the **IP address** of the sending server using sequence numbers.
- PTR and A records in the DNS system establish the **reverse DNS hostname** of the sending server.
- **CSV** and **SPF Classic** examine the HELO hostname of the sending server.
- **SPF Classic** examines the MAIL FROM return-path in the envelope. If a message is undeliverable, this is the identity that gets the bounce message. It requires that forwarders implement Sender Rewriting Scheme (SRS) or its moral equivalent.

The Messaging Anti-Abuse Working Group has sponsored a white paper on the subject. <http://spf.pobox.com/whitepaper.pdf>



- Different authentication tests may be used for different identities. The proposed project will implement multiple authentication schemes and use each one where appropriate. It will also allow local configuration of reputation services. For example, an organization may choose to implement a site-wide reputation policy and query well known third party reputation services at SMTP time. Or an organization may perform only authentication lookups at SMTP time, and delegate the reputation decision to each end-user; end-user MUAs could then ask the addressbook if the sender was recognized.















An MTA actually plays several roles: at a minimum, it operates as a sender and as a receiver of email. The proposed Cheeseplate library will offer functionality to several roles: when invoked by an MTA operating as a receiver of mail, it will perform the logic described above. When sending email, it will sign outgoing messages using the appropriate cryptographic schemes. When operating as a forwarder, it will perform SRS and prepend the headers demanded by Sender ID.

While the codebase is comprehensive, the full vision includes at least one component which the codebase does not cover: publication of SPF records for use by SPF Classic and Sender ID. Getting millions of domain owners to publish records is an education and public relations challenge. Fortunately, advocates of SPF and Sender ID have already organized and executed a grassroots campaign to do this. By some estimates, over 20% of Internet email volume can be usefully tested with SPF. SPF has been around for only about a year, so this accomplishment bodes well for the success of the project.

Mail User Agents (MUAs) also have a part to play. After an MTA has resolved the authentication status of a message, it can further assign a trust rating to the sender based on the sender's reputation, and perhaps mark the message or save it to a different folder. MUAs can also evaluate senders and display messages according to their classification. The project will develop plugins for MUAs to display messages differently based on their rating. MUAs may color-code messages or add simple icons indicating "good" or "bad" status. Forgeries and phishes will acquire a "bad" visual marker, if they are not blocked by an MTA entirely.

Many industry players have endorsed this vision of authentication, reputation, and accreditation. It is the generally accepted road-map for the email industry.

		<b>receiver policy</b>		
		love	shrug	hate
<b>authentication</b>	pass			
	unknown			
	softfail			
	fail			



## STATEMENT OF WORK

THE PRIMARY GOAL is this: the majority of email sites should be able to upgrade their MTAs in the manner to which they are accustomed. After that upgrade, they should be able to turn on sender authentication technologies and immediately reap anti-fraud and anti-phishing benefits. In the case of opensource MTAs, upgrading is cost-free.

We seek to reach an 80% success rate for this goal by May 2006. This means if we target four MTAs and one MUA on four platforms for a total of five patched source distributions and twenty installable packages, we seek to have 20 out of those 25 targets in a completed form by the deadline.

THE SECONDARY GOAL is this: commercial MTA vendors should also integrate that library (or an equivalent implementation thereof) into their products. After the library has been implemented in open-source, we expect to be able to offer assistance to the commercial MTA and MUA vendor industry throughout 2005 and 2006.

These goals minimize the infrastructure burden of change. They do not require that enterprises buy a solution from a single vendor. Nor do they require end users to start doing things very differently.

Rolling out sender authentication is mainly a deployment challenge. Five objectives systematically answer this challenge.

The email ecosystem is extremely heterogeneous. Email sites run a wide variety of operating systems and MTA products. End-users run a wide variety of MUA products. The number of possible combinations is nontrivial. The work therefore comprises five objectives:

1. write a software library, codenamed “Cheeseplate”, that implements a basket of sender authentication technologies
2. patch or provide plugins for a variety of widely used MTA and MUA products to integrate that library
3. shepherd those patches, where possible, into the source distributions of those MTAs and MUAs
4. package the Cheeseplate-enabled MTAs and MUAs for a variety of mainstream operating systems, where possible
5. bundle those packages into the standard distributions of those operating systems, where possible

Modifications will be developed in the form of patches, plugins, or both, depending on the architecture of the software in question. For example, gmail lacks a plugin architecture, so we will produce a patch.

First, multiple authentication specifications must be collected into a single joint standard. Today, the email system is the product of several RFCs. Adding sender authentication to email means adding several more specifications. The Cheeseplate library will implement those additions in a cooperative and standard way.

Second, that library must be introduced into existing email software programs. In the case of opensource software, we can do that directly by patching the code to use the library. In the case of proprietary software, we can offer assistance to commercial developers. There are a large number of MTA and MUA packages on the market. We aim to address as many products as resources permit, prioritizing according to ease of modification and size of userbase.

Third, the patches we develop must find their way into the source distributions of each MTA and MUA. This takes lobbying and persuasion.

Fourth, the MTAs and MUAs must be packaged for convenient installation on a number of common platforms. Again, the choice of platforms depends on ease of modification and size of userbase.

Fifth, these packages must find their way into the standard distributions of those platforms. Again, this takes lobbying.

Software development will follow Agile Development methodologies. Iterative, integrated prototyping will be the norm.

Given the lack of authoritative data regarding marketshare and userbase, some degree of informed speculation is required. Some sources are <http://mailsurvey.os3.nl/> and [http://www.falkotimme.com/projects/survey\\_smtp.php](http://www.falkotimme.com/projects/survey_smtp.php)

<i>Targeted MTAs</i>	<i>Targeted MUAs</i>	<i>Targeted operating systems</i>
Sendmail <sup>1</sup>	Mozilla Thunderbird <sup>1</sup>	Redhat/Fedora <sup>1</sup>
Exim <sup>2</sup>	Outlook+Express <sup>2</sup>	Debian <sup>1</sup>
Qmail <sup>3</sup>	Mail.app <sup>4</sup>	FreeBSD <sup>4</sup>
Postfix <sup>3</sup>	Notes <sup>6</sup>	Solaris <sup>4</sup>
Exchange <sup>6</sup>		Windows <sup>6</sup>

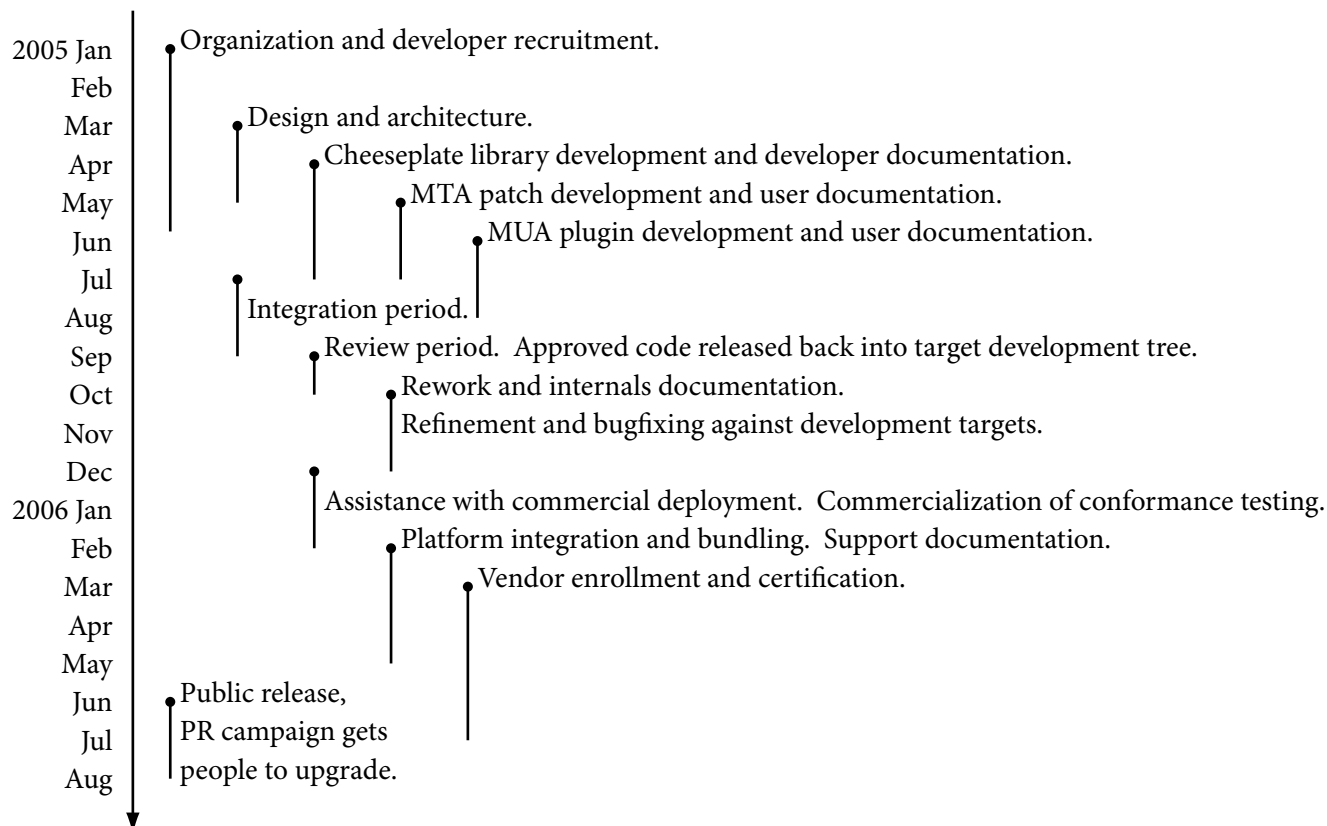
The amount of funding we receive will determine the extent to which we are able to fulfill the above objectives. Targets are listed in order of rough priority. Funding at any of eight levels will deliver usable results. Funded tracks will proceed in parallel.

0	\$150,000	Cheeseplate library only.
1	\$204,500	Sendmail, Thunderbird.
2	\$280,000	Exim, Outlook, Outlook Express.
3	\$350,000	Qmail, Postfix.
4	\$403,000	Solaris and FreeBSD. Mac Mail.
5	\$484,000	Conformance testing.
6	\$577,000	Microsoft Exchange, Lotus Notes.
7	\$750,000	Cost recovery for previous work.

**Intangibles:** This development and deployment experience will be instructive to solving messaging abuse in other media, including Instant Messaging “spim” and mobile (sms/mms) spam and viruses.

## SCHEDULE AND MILESTONES

The following schedule for 2005 and 2006, give or take one or two months, seems achievable:



The following milestones indicate progress on the project.

*All key MTA/MUA developers identified.* We aim to fund those people who are best qualified to write the necessary code. Programmers who are already familiar with the target applications, and distribution maintainers who are responsible for the target platforms, are our first choice.

*Development contracts executed.* Once we have found the right people to do the work, we will execute contracts with them and give them clear direction about what we want them to do.

*Cheeseplate library initial architecture complete.* An initial set of APIs will be developed. A set of stub functions will allow developers to start coding on both sides of the API.

*Inbound HELO checking complete.* The HELO identity is the subject of a number of authentication schemes. Software should perform both reputation and authentication checks on the HELO name.

*Inbound MAIL-FROM checking complete.* The return-path identity is the subject of SPF Classic. Software should perform both reputation and authentication checks on the MAIL FROM value.

*Inbound SUBMITTER checking complete.* The ESMTP SUBMITTER extension is defined in Sender ID. Receiver systems should handle incoming SUBMITTER arguments.

*Outbound SUBMITTER sent where necessary.* In some cases, senders should add a SUBMITTER parameter to the MAIL command.

*Inbound cryptographic checking complete.* We expect to use whatever evolves from DomainKeys and IIM to perform cryptographic authentication of authorship.

*Outbound cryptographic signing complete.* Outbound mail relays are expected to sign messages.

*SRS functionality complete.* Sender Rewriting Scheme is one of the key components of an SPF-compatible system. MTAs will need a new configuration point for SRS.

*Header prepending complete.* Prepending of Resent-\* headers during forwarding is essential to Sender ID.

*Authentication-Results header added.* A new header, “Authentication-Results” has been proposed by M. Kucherawy. MTA software needs to add it.

*Software passes conformance tests.* The conformance testing suite serves as the ultimate test of software readiness.

*MTA Documentation complete.* Each MTA will need its documentation updated to reflect new sender authentication functionality.

*Patches defined for a given MTA source distribution.* Source code patches for a given MTA will integrate added functionality.

*Packages built for a given MTA on a given platform.* Each MTA/platform combination will need an installable package to be built.

*Packages bundled into main distribution for a platform.* Getting the standard MTA package for a given platform to include new functionality can be a major lobbying effort.

*Plugins available for MUAs.* While all this work is going on with MTAs, MUAs will be developing plugins to display Authentication-Results and other tests of the message.

Most of these milestones can be achieved independently. They do not depend on each other. Some of them are prerequisites for others. Milestones will be tracked using standard project management techniques. Some development has already happened on an *ad hoc* basis. These milestones are subject to revision as the landscape of email authentication evolves.

## DELIVERABLES

Where appropriate, these deliverables will be offered on a public website. Software will be released wherever possible as opensource or public domain.

1. The Cheeseplate library will offer the following APIs:
  - a synchronous library to be called directly from code.
  - an asynchronous daemon with the following interfaces:
    - a unix domain socket
    - a TCP socket
    - DNS UDP
    - SOAP
    - REST

The library will be released under an opensource license.

2. we will release patches or plugins to MTAs and MUAs that integrate the Cheeseplate library.
3. we will release patched, ready-to-compile source code versions of MTAs and MUAs that integrate those patches where feasible.
4. we will release easily installable packages for MTAs and MUAs that include the new code, where possible.
5. we will try to get those packages included in mainstream OS distributions. This deliverable can be measured by review of those distributions. These is the primary deliverable by which we aim to measure the success of the project.
6. a standardized conformance suite will include interoperability and unit tests, to ensure that implementations behave as expected. This conformance test may be productized as an interoperability certification program. Commercial MTA vendors can sign up on a website to participate in certification; if they pass the tests, they will get a logo they can display on their products.
7. monthly, quarterly, and annual management reports will discuss the progress of the project and describe how funds were spent.
8. a final report will discuss the success of the project and lessons learned.

## MANAGEMENT PLAN

Many of the sender authentication specifications that are targeted by this proposal have already undergone some degree of development with the backing of corporations and independent citizens. The SPF project, in particular, has had the most momentum and the greatest support to date from the opensource community and the commercial email industry. As the proposed project will be run by a core group of individuals who met on the SPF project, a brief review of SPF is in order.

SPF began around July 2003 under the leadership of Meng Weng Wong of POBOX.COM. Technically, it was a hybrid of two existing proposals, Reverse MX and Designated Mailer Protocol. With an eye to marketability, Meng extracted the best of each proposal and published a specification and reference library. The grassroots responded positively. Thousands of people subscribed to the SPF mailing lists and have, over the past year and a half, helped to bring to its current state. The mailing lists consist of mostly opensource and commercial software developers, system administrators, Internet observers and technology gurus, and representatives of ISPs, banks, and government bodies. Over a dozen individuals now contribute leadership, development, documentation, and publicity.

The proposed project will be produced by the same management team and development community that produced SPF. This group has a deep understanding of the email ecosystem and possesses an unparalleled track record in evaluating, implementing, deploying, and evangelizing sender authentication technologies. Despite being run on a shoestring, the SPF project is very much a going concern. Funds generated by this proposal will therefore constitute a second-round injection of capital. Additional funding will simply take it to the next level.

The combination of RMX and DMP into SPF was the first movement in a theme of synthesis. SPF has since been reused by Microsoft in its proposed Sender ID standard, and may be used as the policy language for other proposals such as DomainKeys and SES.

Integrating all the competing sender authentication schemes into a single, standard library continues that theme. Taking the best of what's available, and using them to complement each other, is inclusive, syncretic, and sensible. Of the available options, not everybody likes everything, but everybody (we hope) will find something they like ... hence the name "Cheeseplate".

Why can't this project be run under the aegis of an existing industry standards body? There are quite a number of them out there.

The Internet Engineering Task Force (IETF), traditionally the home of Internet standards, formed a working group in March 2004 only to dissolve it in October due to lack of consensus. When the debate is framed as "here are half a dozen candidates, let's choose the single best one" it is not surprising that consensus does not emerge. Project Cheeseplate attempts to sidestep this issue by saying "here are half a dozen candidates, let's use the best features of all of them". Besides, the traditional IETF process prefers to recognize an existing *de facto* standard and formalize it *de jure*. It is less able to come up with new standards or to rework existing standards. These were the kinds of reasons that prompted the formation of non-IETF standards bodies such as the World Wide Web Consortium (W3C) and OASIS.

The Messaging Anti-Abuse Working Group (MAAWG) is chartered to assist its members with the evaluation and deployment of anti-spam standards and also to pursue initiatives in education, public policy, and industry collaboration. Its charter does not, however, extend directly to standards or software development.

The Anti-Phishing Working Group (APWG) is likewise focused on discussing and reacting to the phishing problem, but has not, to date, shown any interest in assisting with research, development, and deployment of long-term architectural solutions.

The Federal Trade Commission, as instructed by Congress in the wake of CAN-SPAM, has been following antispam happenings closely, but has expressed a strong preference for industry self-regulation over government regulation.

So it falls to the grassroots to find a way through the thicket. The opensource development community has been a traditional source of effective innovation in the public interest: common examples include Linux, Apache, and MySQL. In the opensource world, the creativity and enthusiasm of amateurs is put to good use, multiple avenues can be explored in parallel, and mistakes are cheap and quickly turn into lessons learned. The SPF community operates in much the same spirit and shares the same organizational dynamics. It harnesses collective action and the spirit of volunteerism to answer the tragedy of the commons and create a public good.

The creation of public goods also falls to the state. ARPA paved the way for the modern Internet. Email evolved under DARPA. It is fitting that HSARPA should help solve spam.

Who exactly are the grassroots? The opensource development model has attracted some of the world's best talent to the project and freed them to contribute according to their unique abilities. These are just a handful of the key contributors to the SPF project. Most, if not all, of these individuals will contribute to the proposed Cheeseplate project. They have already proven their technical competence, good judgement, and ability to work with each other.

**Greg Connor** helps manage and moderate the high-traffic spf mailing lists. He has contributed critical technical insights and balanced opposing viewpoints. In his day job, Greg is a senior system administrator at SGI. In the past, he has also served as Operations Manager for AltaVista, and QA Lead for Apple.

**Dr Phillip Hallam-Baker** is Principal Scientist at Verisign and works very closely with banks and law enforcement to stop phishing and other forms of net crime. A member of the original team that developed the World Wide Web at CERN he has contributed to the design of HTTP and Web Services. He is a recognized expert in the design of Internet security protocols and brings a corporate perspective to standards development and deployment. He will act in an advisory capacity and comment on overall direction.

**Mark Kramer** is an opensource collaborator and a member of the SPF Council. He developed the first Sendmail plugin for SPF and represents the small-enterprise constituency.

**Mark Lentczner** co-authored the SPF specification. He is an expert on language design, standards development, and messaging protocols. He spent many years at Apple as a product manager and later consulted for Openwave, during which time he developed what eventually turned into WML, which is used by hundreds of millions of WAP phones today. He contributes standards-writing expertise, leadership, and public speaking.

**Ben "Shevek" Mankin** developed and maintains the libsrss2 and libspf2 libraries. He is a mathematician specializing in formal security systems and has worked with world-class researchers at the University of Bath. He has many years of experience leading software development on commercial and opensource projects. As a key participant, he will be the Lead Developer of the Cheeseplate library and oversee development efforts for the MTA and MUA tracks in the role of Co-Producer.

**Chuck Mead** was recently elected Chair of the SPF Council, and provides organizational support to the official leadership group. He works at Red Hat Linux on training, is a board member of the Linux Professional Institute, and has a deep understanding of what the free software market wants. He will advise the project representing free software interests and help standardize the products of the project.

The SPF Council is a small, democratically elected body charged with officially representing the SPF community.



**Wayne Schlitt** developed the SPF conformance testing suite and wrote most of the libspf2 C library. He has been involved with the Internet and its precursors for over 25 years. Wayne has been designing multi-user protocols since the late 1970s.

**Theo Schlossnagle**, author of the Ecelerity MTA, was one of the first commercial implementors of SPF and SRS. He has deep experience in mail systems and is a successful entrepreneur in a rapidly growing market. He contributes a deep understanding of the commercial MTA market. As an author of a next-generation MTA, he will advise us on integration issues and high-performance scalability concerns.

**Rand Wacker** is Director of Product Strategy at Sendmail. Sendmail runs approximately 40% of all the MTAs on the Internet. Rand represents that userbase. He contributes a deep understanding of the commercial MTA market. We hope that in addition to advising the project, he will facilitate the Sendmail development effort.

**Meng Weng Wong** founded the SPF project and has been responsible for leadership and strategic direction since day one. He is the public face of SPF: he attends conferences, creates slideshows, meets people, talks to journalists, and writes white papers and grant proposals. He is CTO of pobox.com, an email forwarding service, and developed listbox.com, a mailing list hosting service. He is also Visiting Fellow for antispam at Earthlink and Senior Technical Advisor to the Messaging Anti-Abuse Working Group. His job is to understand what everybody wants, envision a workable future in which all their needs are met, and iterate that vision until everyone accepts it as their own. As a key participant, he will lead the Cheeseplate project in the role of Chief Architect and Executive Producer.

Many other individuals will be involved. We hope to contract with the individuals best placed and best qualified to carry out the work. For example, we will identify those individuals who are already actively involved in developing the targeted opensource products and invite them to assist with this project. This approach efficiently re-uses existing expertise and experience.

## COMMERCIALIZATION PLAN

Typically, commercialization is a problem in bringing innovations to market: specifically, how do we get people to use a technology, and how do we get them to pay for it as well?

Email is free and open. Whatever email evolves into must also be free and open. Many companies have tried to license a proprietary antispam technology to the entire world. While many of these schemes have made many people rich, all of them have failed to end spam for good. The industry recognizes that email is too important for any one entity to control. It is unlikely that any effective, long-term, widely adopted solution to spam will give anyone a monopoly on profits. It is more likely that open standards and open systems, which don't directly make anybody rich, will be more popular.

Reinventing email is like rewiring an old house – except we're not allowed to turn anything off! Whatever antispam technology we come up with, getting people to use it will be a big enough challenge. Asking people to pay for it as well may be asking too much.

So the paradox is that the market won't buy anything that sells. Anything that comes with too bald a profit motive is doomed from the start.

Instead, commercial potential comes from reputation and accreditation systems. Project Cheeseplate, while not making any money in itself, is a necessary enabler for those systems. For more information on the business model there, see the accompanying proposal, *Reputation System Clearinghouse*.

Therefore, this project prioritizes the challenge of getting people to use it above getting people to pay for it. We do technology diffusion in two ways: we directly upgrade opensource software and we help commercial software to follow that lead.

There are, however, opportunities for incidental revenue generation which may be sufficient to keep Project Cheeseplate running even after initial capital is exhausted. Four revenue sources have been identified:

- The commercial MTA market may pay consulting fees to help get their implementations up and running.
- The conformance testing program will charge lab fees.
- The outsourced email sending industry may pay consulting fees to help get their clients properly authenticated..
- Reputation and accreditation services will generate revenue from bread-and-butter contracts. Sender authentication is a prerequisite to those services. They may therefore support Cheeseplate development as an easy way to get into more mailboxes.

## FACILITIES

The work will be performed on the Internet. After an initial face-to-face design meeting, we will collaborate using standard opensource tools such as CVS, IRC, and mailing lists. We will also make free use of tele- and video-conferencing technologies over IP. In particular, we look forward to pair programming with SubEthaEdit under OS X. We may buy some participants new computers to help them work better. We may also rent or buy machines at a colocation facility to act as a development testbed. These facilities will be funded out of the discretionary budget.

<http://www.codingmonkeys.de/SubEthaEdit/>

## GOVERNMENT FURNISHED RESOURCES

This project does not require any special information or data from the Government.

## COST SUMMARY

The bulk of the budget will go toward development labor. The project does not involve any major subcontracts or consumables.

This proposal anticipates that the maximum desired level of funding may not be granted. We define a number of tracks which will still deliver useful functionality. Development of unfunded tracks will still occur to the best of our ability, though it will probably happen more slowly on a hobby basis.

If we are given a budget of \$150,000, we will be able to deliver the Cheeseplate library alone.

With a budget of \$204,500, we can also modify the Sendmail MTA and Thunderbird MUA and attempt to work the improved versions into Debian and Red Hat Linux.

With \$280,000, we can also modify the Exim MTA for Debian and Red Hat and produce plugins for Outlook and Outlook Express.

With \$350,000, we can also modify the Qmail and Postfix MTAs for Debian and Red Hat.

With \$403,000, we can package the abovementioned MTAs for FreeBSD and Solaris as well, and attempt to modify the Mac Mail program under OS X.

With \$484,000, we can develop a comprehensive conformance testing suite, apply those tests to the abovementioned MTAs, and productize the tests as a certification program for commercial MTA vendors and ISPs.

With \$577,000, we can also attempt to develop plugins for Microsoft Exchange and Lotus Notes.

With \$750,000, we can do all of the above and recover some of the costs associated with bringing the SPF project to its current state. The volunteers who have participated in the project and the organizations which have donated resources deserve reimbursement. The majority of these cost-recovery funds will be allocated by decision of the SPF Council, a democratically elected body which represents the SPF community.

If the project completes under budget, excess funds will be left to the discretion of the applicant.

## RESUMES FOR KEY PERSONNEL

Resumes for key members of the management team

- Ben “Shevek” Mankin, architect, project lead, and co-producer
- Meng Weng Wong, architect and executive producer

and a representative sample of opensource collaborators

- Mark Kramer
- Wayne Schlitt

are attached.

These are the kinds of people who write the code that runs the Internet.

## OTHER DHS SUPPORT

None.

EMAIL SENDER AUTHENTICATION  
DEVELOPMENT AND DEPLOYMENT

(PROJECT CHEESEPLATE)

Volume II  
Cost Proposal

pobox.com  
IC Group, Inc.  
mengwong@pobox.com

v1.01 20041217

## COST RESPONSE

Cost Proposal Details for Email Sender Authentication Proposal  
mengwong@pobox.com 20041215 v0.10

running total

						150000	204500	280000	350000	403000	484000	577000	750000
						track 0	track 1	track 2	track 3	track 4	track 5	track 6	track 7
						150000	54500	75500	70000	53000	81000	93000	173000
Management													
project management fee by pobox.com						40000	20000	20000	20000	20000	20000	20000	20000
administrative assistant, part-time						6000	3000	3000	3000	3000	3000	3000	3000
discretionary budget: PR, marketing, website, legal, collaboration facilities, learning travel						12000	8000	8000	8000	8000	8000	8000	8000
						12000	2000	2000	2000	2000	2000	2000	0
Software Development						12000							
						implementation hours	documentation hours	hourly	paired	meeting			
architecture and design						360	planning hours	40	50				20000
cheeseplate library						480		80	80	80			44800
cheeseplate daemon & api						30		10	80	80			3200
conformance testing						360		40	50	100			40000
sendmail mta						120	plugin/patch development hours	20	50	100			14000
							package development, bundling, lobbying						
debian						40		10	50	50			2500
redhat						30		10	50	50			2000
solaris						40		10	50	50			2500
freebsd						20		10	50	50			1500
conformance testing								40	50	50			2000
thunderbird mua						60			50	50			3000
exim mta						80		20	50	100			10000
debian							40	10	50	50			2500
redhat							30	10	50	50			2000
solaris							40	10	50	50			2500
freebsd							20	10	50	50			1500
conformance testing								40	50	50			2000
outlook and outlook express						240		40	50	100			28000
qmail mta						120		20	50	100			14000
debian							40	10	50	50			2500
redhat							30	10	50	50			2000
solaris							40	10	50	50			2500
freebsd							20	10	50	50			1500
conformance testing								40	50	50			2000
postfix mta						120		20	50	100			14000
debian							40	10	50	50			2500
redhat							30	10	50	50			2000
solaris							40	10	50	50			2500
freebsd							20	10	50	50			1500
conformance testing								40	50	50			2000
mac mail.app mua						20		20	50	100			4000
microsoft exchange						240		40	50	100			28000
conformance testing								40	50	50			2000
lotus notes						240		40	50	100			28000
conformance testing								40	50	50			2000
SPF development cost recovery													
pobox.com project hosting								500	100				50000
Disbursement at the SPF Council's discretion								1000	100				100000

This spreadsheet contains a detailed breakdown of labor hours per track. A live copy in the form of an Excel spreadsheet is attached.

ENGINEERING LABOR COSTS are estimated at \$50/hour, with the exception of the Cheeseplate library implementation which is estimated at \$80/hour. Many of the professional programmers on this

project typically charge \$100 to \$240 an hour, but are willing to apply a significant discount because the work is in the public interest.

Each subproject (e.g., library development, target MTA patches, platform integration) will be executed as a subcontract, sometimes with an individual programmer, sometimes with a software development house, and sometimes with an MTA vendor.

PAIR PROGRAMMING is an accepted Extreme Programming methodology that uses two people to write and review code together. This process is understood to produce higher quality code. This is why some of the work items show a doubled hourly wage.

MULTIPLE TRACK APPROACH. This proposal is not structured monolithically. It has been broken out across eight tracks. At the base level of funding, at \$150,000, we can achieve the essential work of producing the Cheeseplate library. At each higher price point we can target more software products on more platforms. Each track can develop in parallel: we will simply engage more developers to address different targets. The nice thing about parallel development is that the project duration remains capped at eighteen months whether two tracks are funded or seven. Due to the project's multitrack structure, we did not budget labor in the traditional "N people for M months" form. Targeted development will instead be subcontracted on a per-project basis. The principal managers of the project will draw an approximate salary based on the amount of work to be done.

THE INITIAL DESIGN MEETING will last three days and bring together developers and architects from all over the world. We estimate the following breakdown per person:

- airfare \$1,000
- a four night hotel stay at \$120 per night for a total of \$480
- a per diem of \$500
- totaling \$2,980 per person

If we bring together eight individuals, the people cost will be \$23,840. Meeting-room facilities are estimated at \$2,000 per day. The total estimated cost for the design meeting is \$33,840. This cost is allocated to Track 0 and distributed among the Travel, Meeting, and Architecture and Design items.

SUBSEQUENT TRAVEL to industry conferences and major adopters to promote the Cheeseplate solution is estimated at \$1,000 per event. These events occur roughly twice a month.

These are a few industry conferences:

- |              |               |
|--------------|---------------|
| • ISPcon     | • Inbox Event |
| • FTC Summit | • IETF        |
| • Usenix     | • MAAWG       |
| • APCAUCE    | • LISA        |
| • OpenGroup  | • APWG        |

**COST SHARE**

None.

**AWARD MECHANISMS**

We request an award in the form of a grant to IC Group, Inc.