次世代 OpenPGP Public Keyserver (OpenPKSD)

鈴木裕信

株式会社 SRA 先端技術研究所

概要

インターネット上で最も利用されている暗号技術は OpenPGP(RFC2440[RFC2440])である。 OpenPGP の公開鍵インフラ(PKI)に使われているサーバソフトが PGP Public Keyserver である。運用は世界各地の鍵管理者コミュニティーのボランティアによって支えられており、インターネットセキュリティの向上に貢献している。

現状で稼動中の PGP Public Keyserver は最新の OpenPGP 規格をサポートできていない。また超大規模な公開鍵データベースとしては処理効率や管理運用面では大きな問題をもっている。新たな PGP Public Keyserver の開発は鍵管理者コミュニティーにおいての課題となっていた。

そこで根本から設計を見直した上で次世代 OpenPGP Public Keyserver である OpenPKSD をスクラッチから実装した。

最終目標は現在世界各地で運用している PGP Public Keyserver を OpenPKSD に置き換えることである。

尚、平成 13 年度作成分 OpenPKSD は openpksd.org サイトにて使用し鍵サーバサービスを提供するとともに、同サイトにてソースコードの配布も行っている。

http://openpksd.org

1. はじめに

1.1 PGP Public Keyserver とは

PGP Public Keyserver とはインターネット上で広く公開されている OpenPGP 仕様の公開鍵パケットの登録および検索サービスのとである。現在、PGP Public Keyserver として広く使われているソフトウェアは Marc Horowitz が作成した pksd[MARC]である。

1.2 背景

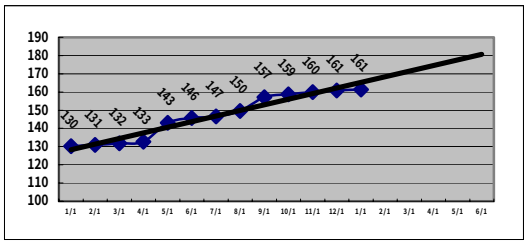
日本で最初の PGP Public Keyserver のサービスは 1994 年 4 月 11 日に筆者が開始した

[ICAT]。ここでの開始とは世界中に分散する PGP Public Keyserver と公開鍵の同期を開始した日である。同期先は MIT の鍵サーバであった。ここから現在の pgp.nic.ad.jp までの 8 年間、日本においてサービスを提供してきた。登録されている公開鍵数であるが、1997 年 12 月 26 日時点では約 5 万 5000 鍵 (pgp.icat.or.jp) であったものが 2002 年 1 月 7 日では約 159 万 5300 鍵(pgp.nic.ad.jp)へと飛躍的に伸びている。

2002 年 9 月から稼動している OpenPKSD サイト openpksd.org 上では 2003 年 3 月末で約 182 万鍵の登録がある。最新の鍵数は http：//openpksd.org のトップページから”Many Keys?”という項目をクリックするとわかる。

PGP Public Keyserver は過去 9 年間のあいだボランティアの手により支えられ続けている。世界各地に散らばる鍵サーバ管理者数名からなるコミュニティーが中心となり各々の持つ鍵サーバ同士を同期させる形でシステムの冗長性を持たせている。また同時にどこでも同じサービスを提供できるようにしている。

登録鍵数は日々増加している。pki.surfnet.nl の鍵サーバ管理者 Teun Nijssen からのメール[TEUN]によれば 2002 年 1 年間で約 31 万鍵増えた。



2002 年 pki.surfnet.nl サイトでの
登録鍵数増加推移：単位は万

現在、多くのサイトで運用しているのは Horowitz 版 pksd であるが、この鍵サーバは数百万という単位の公開鍵をサポートすることは考慮していなかった。非常に良いデザインのため現在はまだ登録、検索に関しては破綻していない。DB として Berkeley DB [SLEEPYCAT] をプログラム内に組込み利用しているので検索に関しては現在でも十分に高速である。

一方で大規模データベース運用としてのメンテナンス性は、あまり良くはない。最も困るのはデータベースを運用しながら、並行してバックアップを取るという機能がない点である。これはデータベースシステムと鍵を検索/登録するサーバプログラムが独立していないためである。

また Horowitz 版 pksd には、このプログラムがリリースされて以降に OpenPGP (PGP) に加わったパケットフォーマットはサポートしていないという問題があった。

本格的な DB エンジンを利用し大量の鍵を扱い、普通の DB ベース運用のような管理でき、最新の OpenPGP のパケットフォーマットをサポートするようなプログラムが必要とされていた。

1.3 期待される効果

期待される効果以下の通りである。

(1) OpenPGP(GnuPG や PGP)を用いた安全な通信のための公開鍵インフラの再構築と世界規模でのサポート

(2) それによる一般ユーザのインターネットセキュリティの健全な発展を促進

(3) 大規模、高負荷、高信頼性ソフトウェアを完全に公開することによるテクノロジートランスファーとその波及効果

(4) 緩やかに結びついている広域分散データベースでのコンテンツ同期のメカニズムの開発とその技術の公開によるテクノロジートランスファーと波及効果

(5) 全世界のフリーソフトウェア・コミュニティーへのエンカレッジとコントリビューション

2. 開発の目標と内容

本テーマにおける開発の目標と内容は以下の項目である。

- 大規模、高負荷耐性、高信頼性に優れた鍵データベース
- OpenPGP(RFC2440)仕様の鍵パケットのサポート
- 世界各地に分散するサーバ間で鍵データベースのコンテンツの同期
- GPL ライセンスの下で公開・配布
- 現在の PGP Public Keyserver の置き換え

2.1 大規模、高負荷耐性、高信頼性に優れた鍵データベース

現時点では運用が面倒だとはいえ Horowitz 版 pksd でも十分に利用できている。しかし、既に Horowitz 版 pksd がデザインされた時とは規模が違っている。このままでは日々増え続ける鍵を管理しきれなくなる日がくるだろう。ここで必要なのは、そのような事態が現実になるのか、あるいはいつなのかという議論ではない。ユーザの観点から何が必要かという議論である。ユーザから見れば将来へのロードマップを示せないシステムなど信頼はしないであろう。その上で技術面からも実用的な「大規模でも大丈夫」「高負荷でも大丈夫」「高い信頼性」を示さなければならない。

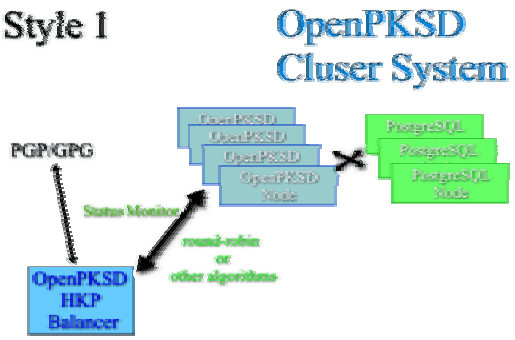
PGP Public Keyserver はネットワークでアクセスされるデータベースの一種である。ソフトウェアの面から大規模化、高負荷耐性、高信頼性を考えた場合、以下のテクニックがあげられる。

（1） クラスタ化されたデータベース
（2） アクセス負荷分散
（3） データベースのノンストップ運用 / 高速なリストア

クラスタ化されたデータベース：平成 13 年度においてはクラスタ化を実現するための調査[WANG]を行った。平成 14 年度はデータベースの分散化は、鍵の ID や Mail アドレス(のハッシュ値)を使い、ストア先のデータベースホストを決めるという Wang 方式をそのまま使った。ここ 2、3 年という範囲ではクラスタ化した PGP Public Keyserver が利用されるとは考えていない。しかし柔軟なクラスタ化が可能であることを示すことは重要である。将来、想像を超える増加があっても対応できるという潜在的能力を示すことができるからである。

アクセス負荷分散：平成 14 年度に hkp ロードバランサーを実装した。分散オブジェクト処理環境 dRuby を用いて鍵パケットの処理を分散化することでクラスタ化を行った。また分散先ノードを決めるためのノード稼動監視デーモンがバックグラウンドで常に稼動しており、鍵パケット処理分散時に適切なノードを決めるようになっている。

バランサーの処理ステップ概略		
Step	Daemon	処理内容
Step 1	バランサー	PGP や GPG から hkp プロトコルで更新、あるいは検索のリクエストが到着する
Step 2	バランサー	hkp プロトコルの parse する
Step 3	バランサー	どのノードを使うかノード監視(Status Monitor)に問い合わせる
Step 4	ノード監視	最も適したノードを返す（ノード監視はすべてのノードの稼動状態を監視している）
Step 5	バランサー	鍵のデータと処理を指定されたノードに引き渡す
Step 6	ノード	鍵を分解(analyze)し、必要な情報を取り出す
Step 7	ノード	鍵の情報(KeyID)から登録先データベースマシンを決め、そのマシンへ登録する



OpenPKSD のクラスタ構成

データベースのノンストップ運用 / 高速なリストア：データベース PostgreSQL [POST] を採用した。PostgreSQL は使いやすく、PC サーバ上では商用 DB 以上の性能を持ち、商用 DB が持つような機能のほとんどを備えている。またデータベースを止めることなく、データーのバックアップが可能である。Vacuum と呼ばれるデータベース内容の“掃除”もできる。商用 DB には見られない PostgreSQL 独自の高速リストア機能はハードウェアをリプレースする時に有用である。実際に約 180 万鍵をもったデータベースを IPA サイトで立ち上げようとしたとき、データベースのリロードおよびインデックス再構築に約 60 分かかっただけであった*(Pentium 4 2.2GHz, Memory 512MB, IDE 40MB/7200rpm)*。

2.2 OpenPGP(RFC2440)仕様の鍵パケットのサポート

過去の pksd のコードを流用することなく openpksd プログラムを Ruby[RUBY]で書くので、OpenPGP(RFC2440)仕様の鍵パケットのサポートではなく、OpenPGP(RFC2440)仕様の鍵パケットをターゲットに作成したと表現するのが正しいかと思われる。Ruby はオブジェクト指向なので、プログラム内部では個々の鍵のパケットをインスタンスとして表現している。そのインスタンス自分自身の振舞(メソッド)を知っている。オブジェクト指向言語を使ったことでプログラム中での鍵の扱いが楽であった。

2.3 世界各地に分散するサーバ間で鍵データベースのコンテンツの同期

登録数が 150 万鍵を超える鍵サーバが、各々自立的に稼働しており、1 つのサイト上で鍵が更新されると同期している他のサイトの鍵サーバが更新される。pgp.nic.ad.jp では 2001 年 1 月 7 日～2002 年 1 月 7 日までの間に鍵の更新同期が 826321 回発生し、新たに約 31 万の公開鍵が追加された[NIC]。

現在の方法は一方向へ鍵が送られる方法である。実際の運用では相手の鍵サーバが止まっていて更新する鍵が送れない、逆に自分の鍵サーバが止まっていて受け取れないなどがある。しかし、下記データを見る限り、特に鍵サーバが停止していないのにも関わらず多くの鍵の違いが発生しているようである（原

因不明)。

各鍵サーバの保持している公開鍵		
鍵数	サーバ名	計測日
1595374	pgp.nic.ad.jp	(2002/1/23)
1605783	pgp.rediris.es	(2002/1/25) [FRAN]
1614164	pki.surfnet.nl	(2002/1/02) [TEUN]
1623342	cc.gatech.edu	(2002/2/01) [PETER]

そこで同期する相手サイトと自分が何を持っていて、何を持っていないのかを知り、必要な鍵を相手から取り寄せる ihave/sendme [RFC850]のような機能が必要になってくる。平成13年度はネットワーク上の2つのサイトにあるデータベースから違いを見つけるアルゴリズムを検討した。

1 つは計算量、もう 1 つはデータ転送量という観点から見た場合、常に有利なアルゴリズムは見つからなかった [KNUTH]。平成 14 年度に行った実装ではデータ転送量を少なくすることを最優先した。これはインターネット帯域の方が、ハードウェアの能力よりもボトルネックになるためである。

Ihave/sendme の処理ステップ概略		
Step	処理側	処理内容
Step 1	両方	KeyID と鍵コンテンツのハッシュ値(SHA1)によるリストを作成する 100 万鍵があった場合､24MB のデータサイズになる
Step 2	サーバ側	作成したリストを相手サーバに送る
Step 3	クライアント側	受け取ったリストと自分のサイトのリスト表と比べ、違うものをリストアップする
Step 4	クライアント側	必要とする KeyID を通知する
Step 7	サーバ側	鍵コンテンツを送付する

3. 本年度の活動状況

平成 14 年度は次の活動を行った。

平成 14 年度活動一覧	
普及	● OpenPKSD サイト openpksd.org にて各種情報の配布 ● OpenPKSD ソースコードパッケージの配布 ● Ruby Conference 2002 (2002/11@seattle)でのデモおよびプレゼンテーション[RUBYCONF] ● IPA Winter での展示
運用	● Openpksd.org サイトによる鍵サービス ● Pgp.nic.ad.jp からの鍵同期
実験	● 大阪市立大学学術情報センター内サーバ pgpsv@media.osaka-cu.ac.jp へのインストール ● 岩手県立大学ソフトウェア情報学部コミュニケーション学内サーバ openpksd.res.soft.iwate-pu.ac.jp へのインストールおよび同期テスト
文章	● OpenPKSD 活用法: クライアント編 (UNIX 編/Windows 編/Macintosh 編) ● OpenPKSD サイト構築: 導入および運用編 (ハードウェア選定/OS 選定/OS インストール/メール設定/PostgreSQL の導入/OpenPKSD の導入と動作確認) ● PGP 公開鍵サーバのデータ同期機能の設計と実装 ● PGP を用いたファイルシステムのセキュリティ強化
実装	● OpenPKSD クラスタ化プロトタイプ実装およびテスト ● Ihave/sendme 実装

4. 成果物

IPA に対する本年度の納品物件は以下の通り。

- 開発成果報告書　1 式
- ソースプログラム　1 式
- ロードモジュール　1 式
- 調査報告書　1 式

本ソフトウェアはフリーソフトウェアとして GNU General Public License を取り入れているためにソースコードとバイナリ実行形式を区別せずアーカイブしている。形式上ソースプログラムとロードモジュールの 2 つを用意して納品としているが、両方に同じものを入れている。尚、ソースコードは頻繁に更新が行なわれているので、もし、利用したい場合は、直接 http://openpksd.org から最新版をダウンロードすることを勧める。

5. 今後の課題

何よりも運用実績を持つ必要がある。運用サイトの中心として openpksd.org は出来るだけ長く運営していきたい。またバグが発生した場合に迅速に対処する必要があるし、こまめなリリースも必要であろう。

鍵サーバがたとえ Yahoo や eBay のような人気サイトになってアクセスが激増したとしても処理できると思われる。しかし、実装のテストは最大 5 ノードまでしか行っていないし、負荷テストも 2 台を負荷発生装置として 3 台をノードにする形で行うという小規模なものに終わったのみである。この点についてもう少し突っ込んだ実験が必要である。

ノードマシン

現実の運用の面からは、故意の利用妨害(DoS)に対する耐性を高めなければいけないだろう。尋常ではない頻度で繰り返しアクセスしてくるアドレスを規制したり、流量をチョークしたりといった対応などが考えられる。現在は単純に単位時間内のアクセスを IP アドレスで見て、多い場合は遅滞を加えるという形にしているが、もっとインテリジェンスな方法を考えたい。

同時に鍵サーバにはアクセスコントロールは存在しないのでアクセスコントロールも考えたい。Trusted ユーザのみが鍵を更新/参照できるといったサービス・実装の拡張もニーズがあるので、今後、このような機能も加えていきたい。大量の公開鍵がデータベース化しているので、ある公開鍵がどの程度信頼できるものかといった評価などのサービスも考えられる。

現在、有志の手による PGP Public Keyserver の RFC Draft が作成され、議論している最中である。この中には、現在のいずれの PGP Public Keyserver には実装されていない新しい仕様などが入っているので、これを実装することも視野に入れている。

6. まとめ

インターネット上での暗号セキュリティツールの主役は PGP であり GNUPG である。OpenPKSD はあくまでも裏方のインフラを受け持つ。今回のプロジェクトは長年使われているインフラの改良と整備というプラクティカルな活動だと位置付けることができるだろう。

pksd の後継としての地位を築いたとしても、OpenPKSD のソフトウェアをインストールするサイトは全世界で 100 サイト以下であろう。ただし OpenPKSD のサービスを利用する者は登録鍵のオーダー、つまり 100 万単位の利用者ということになる。運用側として

は、やはり長期に安定して使えるプラットフォームをユーザにいかに提供するかが課題となる。それを提供するのも OpenPKSD の役目である。

このまま順調に開発が進めば Horowitz 版 pksd と世代交替できる可能性は高まるだろう。

7. Acknowledgement

OpenPGP 及び関連ツール等に関しての調査、ドキュメント、OpenPKSD サイト構築は岩手県立大学ソフトウェア情報学部村山優子を中心とした PGP-folk のメンバーである山根 信二、王 家宏、瀬川 典久、市澤 浩史、藤原 光照、高橋 充、井上 智貴、影山 智徳、中野 隆浩、加藤 由恵、井上 達也、庄司 和宏、間宮 章彦、鈴村 圭史、瀬戸 綾介、宮沢牧子、斎藤 歩、権藤広海、荒川健介、市川快、秋山和隆、手塚一郎によって行われた。

同期アルゴリズム等は大阪市立大学学術情報総合センター中野秀男を中心とした大西克実、石橋勇人らとのメーリングリストでのディスカッションをベースにまとめたものである。また同大学での OpenPKSD サーバ構築も 3 名の協力によって行われた。

PGP Public Keyserver の情報を調べるに当たり Peter N. Wan (ジョージア州工科大学)、Francisco Jesus Monserrat Coll (RedIRIS)、Teun Nijssen (Surfnet)の協力があった。

IPA 中村滋の協力により他サイトへのデータベースコピーのテストが行われた。

OpenPKSD プロジェクトには SRA-KTL 馬場尚子、桝田義一が参加した。(文中敬称略)

8. 関連資料

[RFC2440] http://www.ietf.org/rfc/rfc2440.txt

[MARC] http://www.mit.edu/people/marc/pks/

[ICAT] http://pgp.nic.ad.jp/docs/icat_rep_1.txt

[TEUN] Teun Nijssen *Teun.Nijssen@kub.nl* との私的メール交換 (2002/1/27)

[SLEEPYCAT] http://www.sleepycat.com/

[WANG] http://openpksd.org/docs/report2001/PGP-005/PostgreSQLandCluster.pdf

[POST] http://www.postgresql.org/

[RUBY] http://www.ruby-lang.org/en/index.html

[RUBYCONF] http://www.zenspider.com/Languages/Ruby/RubyConf2002/index.html

[NIC] http://pgp.nic.ad.jp/docs/2001.txt

[FRAN] Francisco Jesus Monserrat Coll *francisco.monserrat@rediris.es* との私的メール交換 (2002/1/29)

[PETER] Peter N. Wan *Peter.Wan@cc.gatech.edu* との私的メール交換 (2002/2/3)

[RFC850] http://www.ietf.org/rfc/rfc850.txt

[KNUTH] Donald, E. Knuth, *"The Art of Computer Programming"*, 2nd ed, vol 3, 1997.

[OPENPKSD] http://openpksd.org