Secure Communications During the End Game

How can one small committee communicate securely, across an unsecured medium, monitored by a hostile adversary with almost unlimited resources, with as many pilots who care to engage? How can the thousands of pilots know that the message communicated on an unsecured network has not been forged?

It would be easy enough for the airlines, or their enforcers in various government agencies, to simply trespass (hack) the internet domain playing host to the communications of OPERATION ORANGE, and disrupt the entire effort. They could entice sympathetic media to broadcast contrary messages, claiming they were authentic, and send OPERATION ORANGE into chaos. It certainly would take little effort to hire people, claiming to be members of The Committee, in order to confuse everyone and disrupt the operation.

To keep everything moving in our favor, we need to overcome a few obstacles, and the bulk of these obstacles can be overcome by using public key cryptography and a few related functions.

Here is a walk-through of how the communications will likely transpire.

The SELF-DECRYPTING ARCHIVE (SDA)

When we have all the necessary components to carry out the SOS/protest, we will come out with a video on YOUTUBE and a companion entry on the OPERATION ORANGE websites. The website entry will have a link to a .zip file containing a few “SELF-DECRYPTING ARCHIVES.” The SDAs will contain various messages we can reference after the SOS has been announced. This gets all the messages we need into your hands before the operation even starts.

A SDA is a very useful tool in this application because it carries a very strong encryption medium, but **does not** require the recipient be a user of the host software (“Pretty Good Privacy” aka “PGP”). The SDA comes with everything you need to decrypt the archive, except the operating pass phrase. Anyone with the SDA file, off-the-shelf computer, and the pass phrase can read the contents. This is key because most pilots participating in the SOS/protest will not go through the effort to download the PGP trial ware from Symantec. It doesn’t matter with SDAs.

We will have several SDAs - one of which will contain the “return to work” message. A few others will contain messages covering other contingencies, and some will be decoys.

This puts us in a position where we only have to communicate a “pass phrase” to direct the operation. We will still be able to operate without our own website. We could disseminate the pass phrase via any network - our website, media outlet, YOUTUBE, government, airlines, pilot associations, Twitter, etc.

PUBLIC KEY CRPTOGRAPHY

We still have the problem of imposter SDAs distributed in our name. After all, one SDA looks like another. This is solved by using the "authentication/verification" functions of public key cryptography. As we explain in our "AUTHENTICATION" document, located in the masthead menu on our OPERATION ORANGE websites, we can use the signature function of our unique public/private key pairs we have generated, to digitally sign the SDAs we create. This prevents a third party from substituting a faulty key pair and trying to pass off a counterfeit SDA to the pilots.

Each PGP user creates a "key pair" when they initially install the PGP program. This gives each user a "private" key that they keep to themselves and a mathematically corresponding "public" key that is supposed to be widely distributed over unsecured media. Wide dissemination of public keys does not compromise the integrity of the encryption or authentication. In fact, the more widely distributed the public key, the more difficult it is to impersonate.

Private keys are used to sign documents and the corresponding public key is used to authenticate that signature. The public key is not capable of signing a document; it can only verify a signature from its corresponding private key. If the public key shows "BAD SIGNATURE," either the document has been altered, the document has been signed by an imposter private key, or the public key is an imposter. As long as you have the proper public key (and are running PGP), you can verify any document signed with the corresponding private key.

What prevents someone from impersonating a public key?

Each public key comes with a "digital fingerprint," which is a series of numbers, letters, and phonetically distinct words unique to the particular mathematical arrangement of the public key. We have listed all four of our keys in the "AUTHENTICATION" document in the masthead menu on our OPERATION ORANGE websites, along with their digital fingerprints. You can simply download the keys, click on "KEY PROPERTIES" and verify the signature with the ones we have been providing since OPERATION ORANGE was in its infancy.

As an additional layer of security, public keys can be digitally signed by other private keys, and verified by the corresponding public keys in the same manner as digitally signed documents. All four of our keys have been digitally signed by the other members of The Committee. If anyone is to impersonate our keys, they would lack the other three signatures. This is why we published our keys well in advance of OPERATION ORANGE gaining any significant exposure.

We also have published an “OPERATION ORANGE REVOKER” key. In the unlikely event one of our keys or committee members is compromised, we can publicly “revoke” the corresponding key by using the revoking function of the OPERATION ORANGE REVOKER. We simply republish the key on the global key server or our own website with a revoked status. We can then reissue a new key with all the authentication precautions of the original key. Please refer to the PGP users guide for further information.

We have published our keys on our websites and they are also available on the PGP Global Key Server, kept by PGP/Symantec. Please feel free to go to these sites and verify that the keys you have are authentic.

We would use the “private” key to uniquely sign the SDAs and .ZIP files, as we have done in the DOCUMENTS AND SIGNATURES files we have on the masthead menu on our OPERATION ORANGE websites. Those pilots who are operating PGP on their own computers can then take our freely disseminated “public” keys and verify the signatures that accompany the various SDAs and .ZIP files.

If one byte of the file has been altered, no matter how trivial, the hash on the signature will not match the document and PGP will flag the file as “BAD SIGNATURE.”

Not everyone needs to run PGP. We only need enough people to run PGP to be able to create enough noise to let everyone know the files are not genuine. The more pilots running PGP, the more secure the operation.

Symantec has PGP Trial software and we have detailed how to go about getting it. It is free, but you need to register with Symantec to download it. The Trial software contains all the functionality needed to fully authenticate documents in OPERATION ORANGE, even after the full functionality trial ends. It is good software and worth every penny, should you elect to purchase a license. PGP has long been considered the gold standard of encryption software available for commercial use. It is so powerful, the US government declared it a munition and attempted to prosecute the creator of the software for distributing it outside the United States. He beat the charge by publishing the code under the PROTECTION OF THE FIRST AMENDMENT.

It has been said that there are two kinds of cryptography in this world: cryptography that will stop your kid sister from reading your files, and cryptography that will stop major governments from reading your files. PGP is the latter sort of cryptography.

At this point in the scenario, a SOS date has been chosen, a .ZIP file containing several SDAs and digital signatures has been distributed, and many have installed PGP and have verified the documents with the public keys. The ATA/government can't forge a SDA or key, and all the necessary communications are already in the hands of the pilots planning to SOS 6-8 weeks subsequent. What next?

In the event The Committee is compromised, we still need to be free to turn off the SOS. By sending out the SDAs, the SOS becomes "FAIL ACTIVE," meaning that unless you hear otherwise, the SOS goes as scheduled. The only way to turn off the SOS is for The Committee to release the pass phrase to unlock the "return to work" SDA.

When the actual SOS date and time approach, we will not need to give an additional instruction for everyone to "ORANGE-OUT." It is presumed the original communication that sets the date-time for the protest contains everything everyone needs to know about how and when to "ORANGE-OUT." The operational security benefits of this arrangement are obvious.

AN EXAMPLE COMMUNICATION

Here are an assortment of SDAs and keys for us to use to demonstrate all that we have been discussing. **THESE SDAs AND KEYS ARE NOT TO BE USED FOR THE ACTUAL SOS.** They are purposefully frivolous so as to reduce any of the confusion that is going to surround the events of the SOS. Please discard all these keys, SDAs, and decrypted messages after you are comfortable with the lesson objectives.

This tutorial is designed to give you a basic outline of how to use PGP's authentication functions. At the conclusion of the tutorial, you should be able to have a working understanding of:

-What a "public" key is and its function in the authentication process
-What a "key pair" or "private" key is and its function in the authentication process
-How to activate a Self Decrypting Archive
-How to import public keys to your keyring
-How to locate the ID of a public key
-How to locate the digital fingerprint of a public key
-How to locate the biometric fingerprint of a public key
-How to sign/verify a public key
-The difference between an unverified key and a verified key
-How to check the verification of a signature file
-What information is displayed on a valid signature
-The components of a valid SDA from OPERATION ORANGE
-The difference between a signed SDA and an unsigned SDA
-What a "Bad Signature" is, and how it is displayed in PGP
-What is displayed when an unknown key is used to sign a file.

Download the following .zip file.

Go to the Symantec website and download PGP Whole Disc Encryption. You will need to register with Symantec. It is free. Install the software per the instructions.

The downloaded .zip file should contain:

- 6 Self Decrypting Archives
- 5 Signature files corresponding to 5 of the SDAs
- 2 “authentic” public keys
- 1 folder containing 2 “imposter” public keys

Save these files to your desktop.

Select both of the two “authentic” public keys

Right click on the keys

Select “PGP Desktop” → Import Keys

Import both keys to your PGP keyring via the dialogue (Acorn and Pinecone). Double-click on Pinecone to bring up the Key Properties display.

The digital fingerprint of Pinecone (ID: 0x7E135B1F) is: (Hexadecimal)
A46E F56B 1ADC B4F0 78E1 2394 5551 2A38 7E13 5B1F

(Biometric)
regain headwaters vapor Hamilton beehive sympathy scenic upcoming island tolerance blowtorch molecule edict enchanting brickyard consulting locale barbecue erase businessman

Double-click on Acorn to bring up the Key Properties display.

The digital fingerprint of Acorn (ID: 0x3C09C9BB) is:
(Hexadecimal)
EFF6 7E56 2CCF E738 FD4D 2E2D 3A84 C31D 3C09 C9BB

(Biometric)
uncut vocalist locale escapade Burbank Saturday transit consulting willow disruptive buzzard clergyman cleanup Jupiter snowcap breakaway cobra applicant spearhead publisher

Verify Pinecone’s fingerprint and key ID number. Once you have verified Pinecone’s ID, sign the key with the private key you created when you installed PGP.

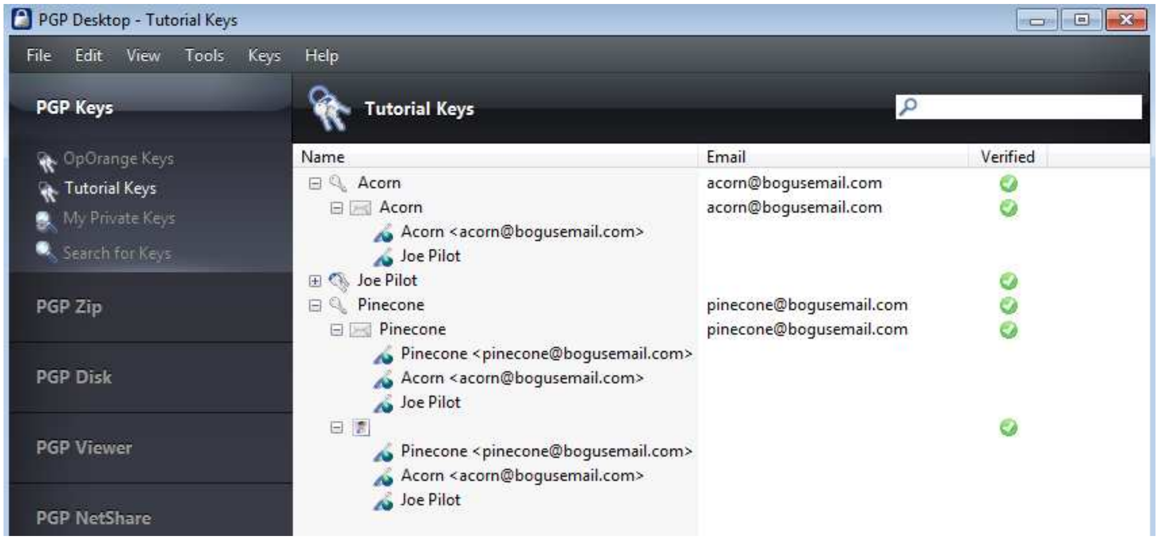
To sign a key,
-Right click on the key you wish to sign
-Select SIGN from the menu
-Verify the elements of the key you wish to sign, including the digital fingerprint
-Click OK
-Select your private key from the dropdown menu
-Enter your pass phrase for your private key
-Click OK

Verify Acorn’s fingerprint and key ID number. Once you have verified Acorn’s ID, sign the key with the private key you created when you installed PGP.

This will change the “VERIFY” status to VERIFIED and remove the question marked box superimposed over the photo in the Acorn public KEY PROPERTIES display.

You will note that Pinecone is signed by Acorn, as Acorn has also verified Pinecone’s digital fingerprint. Thus, in order to impersonate Pinecone’s key, one would have to also impersonate Acorn’s key, or have the owner of Acorn sign a key they knew to be false. Click on the expansion box for a public key on your keyring and the various emails, photos, and keys are shown with all the signatures known to your keyring. If they are signed by an “unknown key,” do not be alarmed. A key may be signed by a public key you do not posses. You just can not rely on such a signature, but it doesn’t invalidate the key.

At this point your KEYRING display should look something like this:



Name	Email	Verified
Acorn	acorn@bogusemail.com	✓
Acorn	acorn@bogusemail.com	✓
Acorn <acorn@bogusemail.com>		
Joe Pilot		
Joe Pilot		✓
Pinecone	pinecone@bogusemail.com	✓
Pinecone	pinecone@bogusemail.com	✓
Pinecone <pinecone@bogusemail.com>		
Acorn <acorn@bogusemail.com>		
Joe Pilot		
		✓
Pinecone <pinecone@bogusemail.com>		
Acorn <acorn@bogusemail.com>		
Joe Pilot		

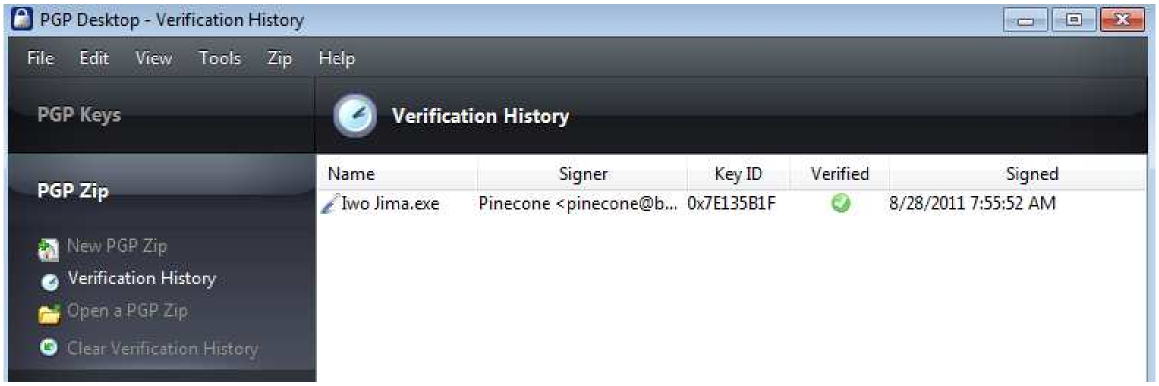
You have the two "authentic" keys called Pinecone and Acorn. Pinecone is the primary key used for authentication and Acorn is a key used to sign Pinecone. An authentic Pinecone key would have both Pinecone's fingerprint as well as being signed by Acorn. You verified both keys as genuine and then signed with your personal key.

Note: Your keyring will have your personal key in the place of Joe Pilot. Acorn and Pinecone will not be signed by Joe Pilot, but YOUR key. You may have other keys on your keyring as "public" keys if you downloaded our keys from the PGP Global Keyserver or the OPERATION ORANGE websites. **In this tutorial ONLY, every place you see "Joe Pilot," you should expect to see your key.**

We are all set up and ready to go through the various scenarios for genuine and imposter SDAs.

First Scenario - Genuine SDA

Double click on the signature file called "Iwo Jima." This should bring up your PGP program and the VERIFICATION HISTORY screen. The IWO JIMA SDA has been signed by Pinecone; it gives Pinecone's key ID, shows that the signature is verified against a valid public key you hold on your keyring, and gives the date/time it was signed.



Name	Signer	Key ID	Verified	Signed
Iwo Jima.exe	Pinecone <pinecone@b...	0x7E135B1F	✓	8/28/2011 7:55:52 AM

This shows that the file associated with the signature is the exact file signed with the private key and that it has not been substituted nor altered in any way.

Now that we know "IWO JIMA" SDA is a valid SDA, let's see what's inside.

Double-click on the Iwo Jima SDA.

A pass phrase entry box appears. The pass phrase for this SDA is:

tangerine

Enter the destination you wish for the contents of the SDA in the output directory window. Enter the pass phrase. Click OK.

Two files appeared in the directory you selected. You should have:

- "DMO" PDF file
- DMO.pdf signature file

Verify the DMO signature file in the same way you verified the Iwo Jima signature file. Double-click on the signature file and see how PGP displays it in your VERIFICATION HISTORY window.

PGP should show it as a valid signature with all the same information it previously displayed when you verified IWO JIMA. You now know the DMO file is uncorrupted. This is an extra security measure we will incorporate into any valid SDA we produce. We will presume the documents from the SDAs will be widely distributed when the SOS concludes and we want to be certain the documents are uncorrupted and verifiable.

Open the DMO file and read the contents. It is a silly phrase to prevent this from being confused with documents we will circulate during more serious times. These files are only tutorials and should be deleted after you are comfortable with the object lessons.

That is how a valid SDA will present itself:

- Valid signature from one or more OPERATION ORANGE keys.
- Passphrase that unlocks the SDA
- Valid signature of the internal document from one or more of the OPERATION ORANGE keys. The signature may not necessarily be the same signature that verified the SDA. It is important you download all 4 of the OPERATION ORANGE keys.

SECOND SCENARIO - Unsigned SDA.

The most common way to attack the method of communicating we have chosen would be to introduce SDAs that we did not originate. It will likely be unsigned because most pilots will not be running PGP on their computers. This is how such an imposter SDA would look:

Double-click on the Tinian SDA.

A pass phrase entry box appears. The pass phrase for this SDA is:

pumpkin

Enter the destination you wish for the contents of the SDA in the output directory window. Enter the pass phrase. Click OK.

One file appeared in the directory you selected. You should have:

- “111“ PDF file

Note that there was no verification signature for either the SDA or the internal file. This is one method the opponents of OPERATION ORANGE could use to thwart the operation. They could do something like this to send out contrary messages.

The 111 file is a bogus file that came from an unverified SDA.

SCENARIO THREE - SDA With Bad Signature File

Rather than just sending out a SDA without an accompanying signature, the opponents could take the existing signature file and attach another SDA. This won’t pass the signature verification, even though they are using the signature that was originally issued with the original SDA. The signature file that we created won’t match the bogus SDA and will show as “BAD SIGNATURE.” Remember, if one byte of the file associated with the signature file has been changed, the signature no longer matches and will show as a bad signature.

Simulate that you received the Guadalcanal SDA and signature files along with the pass phrase:

carrot

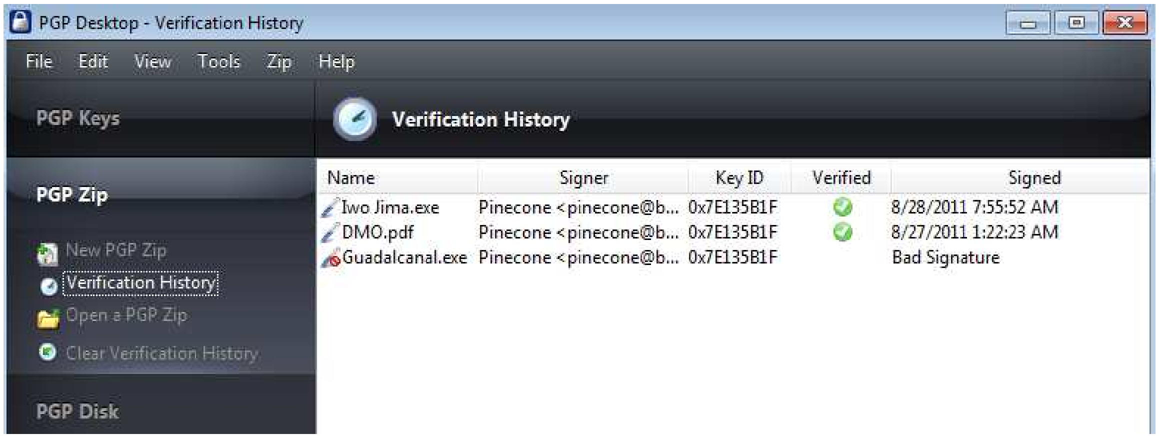
If you do not verify the signature file and immediately open the SDA, you will get a message that was not part of the original SDA, because the SDA you have (Guadalcanal) was substituted.

Open the SDA by double-clicking on it. Enter the destination directory and pass phrase. You should have one file:

- “222” PDF

This is what you would get if you didn’t verify the signatures. Note that the internal components of the SDA did not include a signature file for the document. Our documents will contain a signature file.

Go back and double click on the Guadalcanal signature file. PGP will flag this as a “Bad Signature.”



Name	Signer	Key ID	Verified	Signed
Iwo Jima.exe	Pinecone <pinecone@b...	0x7E135B1F	✓	8/28/2011 7:55:52 AM
DMO.pdf	Pinecone <pinecone@b...	0x7E135B1F	✓	8/27/2011 1:22:23 AM
Guadalcanal.exe	Pinecone <pinecone@b...	0x7E135B1F		Bad Signature

Guadalcanal.exe is shown to be not the same file as the signature, even though the signature comes from a valid public key on your keyring. It lacks the VERIFIED check and shows “Bad Signature” in place of the date/time it was signed.

Always verify signatures.

FOURTH SCENARIO - Signed by Unknown Key

You should **not have** downloaded the “imposter” keys at this point. You should only have the “authentic” keys (Pinecone and Acorn) on your keyring. If you have the imposter keys on your keyring, you will not be able to complete this scenario.

Double-click on the Midway SDA. The pass phrase for Midway is:

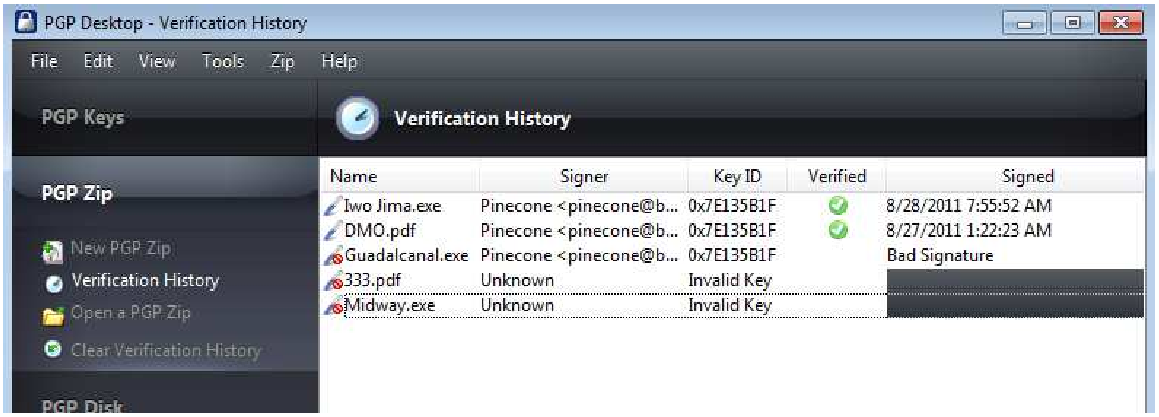
The General Lee

Note that all PGP pass phrases are case sensitive. Enter the destination directory and pass phrase. You should have two files:

- “333” PDF

- 333.PDF signature file

If you attempt to verify the signature for the 333.PDF or the Midway SDA, PGP will show that the signer is unknown and the key is invalid.



Name	Signer	Key ID	Verified	Signed
Iwo Jima.exe	Pinecone <pinecone@b...	0x7E135B1F	✓	8/28/2011 7:55:52 AM
DMO.pdf	Pinecone <pinecone@b...	0x7E135B1F	✓	8/27/2011 1:22:23 AM
Guadalcanal.exe	Pinecone <pinecone@b...	0x7E135B1F		Bad Signature
333.pdf	Unknown	Invalid Key		
Midway.exe	Unknown	Invalid Key		

This is what happens if someone signs a document with a key that doesn't correspond with a public key on your keyring.

SCENARIO FIVE - SDA Signed by Imposter Key

There is one more method someone can use to attempt to pass off an imposter document as genuine. If they can get you to import their public key and verify it, you won't have any idea that you are getting documents from a source you don't trust.

This can only happen if you are careless with the importation and verification of keys, which is why doing a fingerprint verification is so important. Each key is unique, and each key carries a unique digital fingerprint. These features can't be forged with our current understanding of mathematics.

Let us say someone interested in forging our communications put keys on the PGP Global Keyserver, or the OPERATION ORANGE downloads, which appeared to be identical to our keys. This is done with the intention of deceiving you into importing their keys in lieu of our keys. How this could happen is a matter of conjecture, but for the purposes of this tutorial, let's simulate that it happened.

Browse to the “imposter keys” in the .zip file you downloaded.

Highlight both keys (pinecone and acorn)

Right click on the keys
Select PGP Desktop →Import Keys

Select “Import” for each key. This saves unverified public keys to your keyring. Note that the “VERIFIED” column on the PGP Keyring display shows dashes surrounded by grey circles. This denotes a public key which has not been verified, and is perfectly normal when importing keys. You need to sign the new keys with your private key in order to change the key to “VERIFIED,” which is denoted by a checkmark surrounded by a green circle.

This is normally where you would verify the digital ID and fingerprint of the public key against a known ID or fingerprint. For purposes of this exercise, we gave you the ID and fingerprint when we asked you to download the “authentic” keys (Pinecone and Acorn), just as in the real OPERATION ORANGE keys, we have given these ID/fingerprints in the AUTHENTICATION document on the masthead menu on the OPERATION ORANGE websites.

Note: to minimize confusion in this tutorial, the “authentic” keys are named “Pinecone” and “Acorn,” whereas the “imposter” keys are named “pinecone” and “acorn.” The first character of the key name is capitalized in the authentic key, and lower cased in the imposter keys. In a real-world scenario, you will likely encounter the key name being identical in every way. For purposes of this tutorial, we will overlook the capitalization distinction and simulate the key names are identical.

For this tutorial, simulate that you did not verify the digital fingerprints of the imposter keys, as this would be a necessary oversight to import imposter keys. You can see that the key ID/fingerprints are different, even though the picture and name are the same on pinecone, and it appears to be signed by the same key (acorn).

Do not sign the new keys at this time. We will discover what an unverified/invalid key looks like.

Double-click on the Midway signature file. PGP will find the corresponding public key on your keyring (pinecone), display the keyname, ID, and verification status. In this case, the verification status is “invalid,” because it was not signed by your private key. This is to warn that, while you do have the public key, you have not verified it as genuine. The signature status shows the date/time it was signed, but flags it as an “INVALID KEY.”

Sign (verify) the two imposter keys with your private key.

This changes the verification status to “VERIFIED.”

Double-click on the Midway signature file (last time).

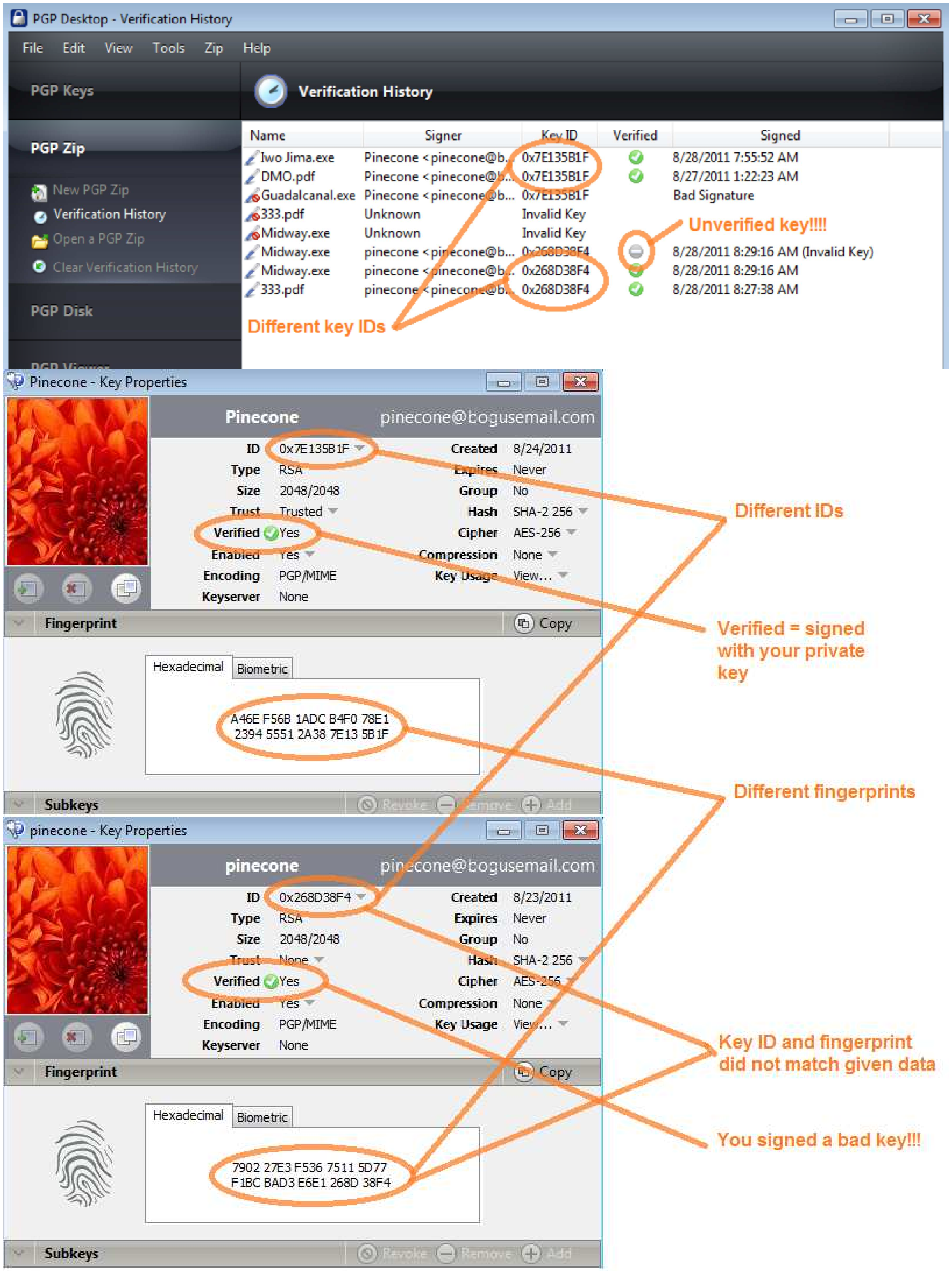
PGP will find the corresponding public key on your keyring (pinecone), display the keyname, ID, and verification status. In this case, the verification status is now “valid,” because it has been signed by your private key.

Double-click on the Midway SDA. Enter the destination you wish for the contents of the SDA in the output directory window. Enter the pass phrase (The General Lee). Click OK. (You may have to authorize the two files to overwrite two existing files from the previous exercise.)

Double-click on the 333.PDF signature file.

PGP will find the corresponding public key on your keyring (pinecone), display the keyname, ID, and verification status.

Notice that the signature is considered “VALID,” but only because it is referencing an imposter key. The imposter key was only possible due to not verifying the digital fingerprint and key ID.



Verification History

Name	Signer	Key ID	Verified	Signed
Iwo Jima.exe	Pinecone <pinecone@b...	0x7E135B1F	✓	8/28/2011 7:55:52 AM
DMO.pdf	Pinecone <pinecone@b...	0x7E135B1F	✓	8/27/2011 1:22:23 AM
Guadalcanal.exe	Pinecone <pinecone@b...	0x7E135B1F		Bad Signature
333.pdf	Unknown	Invalid Key		
Midway.exe	Unknown	Invalid Key		
Midway.exe	pinecone <pinecone@b...	0x268D38F4	⊖	8/28/2011 8:29:16 AM (Invalid Key)
Midway.exe	pinecone <pinecone@b...	0x268D38F4	✓	8/28/2011 8:29:16 AM
333.pdf	pinecone <pinecone@b...	0x268D38F4	✓	8/28/2011 8:27:38 AM

Unverified key!!!!

Different key IDs

Pinecone - Key Properties

Pinecone pinecone@bogusemail.com

ID	0x7E135B1F	Created	8/24/2011
Type	RSA	Expires	Never
Size	2048/2048	Group	No
Trust	Trusted	Hash	SHA-2 256
Verified	✓ Yes	Cipher	AES-256
Enabled	Yes	Compression	None
Encoding	PGP/MIME	Key Usage	View...
Keyserver	None		

A46E F56B 1ADC B4F0 78E1
2394 5551 2A38 7E13 5B1F

Different IDs

Verified = signed with your private key

Different fingerprints

pinecone - Key Properties

pinecone pinecone@bogusemail.com

ID	0x268D38F4	Created	8/23/2011
Type	RSA	Expires	Never
Size	2048/2048	Group	No
Trust	None	Hash	SHA-2 256
Verified	✓ Yes	Cipher	AES-256
Enabled	Yes	Compression	None
Encoding	PGP/MIME	Key Usage	View...
Keyserver	None		

7902 27E3 F536 7511 5D77
F1BC BAD3 E6E1 268D 38F4

Key ID and fingerprint did not match given data

You signed a bad key!!!

Notice that we did not decrypt all the SDAs. Tarawa and Okinawa were valid SDAs, but are either decoys or contingency instructions. You can verify them via the signature files, but you can't decrypt them unless you have the pass phrase. The pass phrases we will use in the real-world operation will be substantially more complex than the ones we used for this tutorial.

You should have a working understanding of the necessary skills for verification of our SDAs, or any PGP signed file. We certainly hope the all these precautions are an over preparation and that all goes well. Preparation is the prelude to success, so please work through these exercises until you are comfortable with all the enabling objectives.

These tools allow us to send uncorrupted messages to the participating pilots. Neither the ATA nor government can stop this or impersonate our signatures. This technology was not available to the pilots that went before us, so let's use it to our advantage.

Please consider downloading the trial version of PGP. The more pilots we have using the software, the easier the end game will be.

Until that time, please spread the word. Pressure your flying partners, your union leadership, and your buddies at other airlines. This is the way we take back our profession.

The rules of the game have changed. They changed a long time ago but we have not adjusted our tactics. It is time to update our tactics and write our own rules for a change.

It is our turn to lead.

THIS IS OUR TIME.

Links For This Document

(PGP User’s Guide)
http://operationorange.org/crypto.pdf

(PGP Global Keyserver)
https://keyserver.pgp.com/

(Master Documents and Signatures)
http://operationorange.org/masterdocs.zip

(Authentication Document)
http://operationorange.org/authentication.pdf

(End Game Training Files)
http://operationorange.org/endgametrainingfiles.zip

(PGP Trial Software)
http://www.symantec.com/business/whole-disk-encryption

To get the PGP Trial version, go to the Symantec website, and follow the following to get to the trial version of PGP Whole Disk Encryption:

- - -Select "Business" from the top masthead
- - -Select "Products" - "Products A-Z"
- - -Scroll down and select "Whole Disk Encryption"
- - -Click on "Trialware"
- - -Click on "PGP Whole Disk Encryption Trialware"

Follow the directions for download. Site registration is required. Download is free after registration. This will enable you to receive the trialware licensing code. Note that some functionality will disable after the trial period, but the "keys" function will remain active after the trial is over.