

IEEE Standard for Biometric Open Protocol

IEEE Communications Society

Sponsored by the
Standards Development Board

and the

IEEE Technical Activities Board

Sponsored by the
Technical Committee on COM/SDB



IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 2410™-2017
(Revision of
IEEE 2410-2015)

IEEE Standard for Biometric Open Protocol

Sponsor

Standards Development Board
of the
IEEE Communications Society

and the

Technical Committee on COM/SDB
of the
IEEE Technical Activities Board

Approved 28 September 2017

IEEE-SA Standards Board

This is a Redline Document produced by Techstreet, a business of Clarivate Analytics

This document is intended to provide users with an indication of changes from one edition to the next. It includes a full-text version of the new document, plus an indication of changes from the previous version.

Redlines are designed to save time and improve efficiencies by using the latest software technology to find and highlight document changes. More professionals are using valuable new technologies like redlines, to help improve outcomes in a fastpaced global business world.

Because it may not be technically possible to capture all changes accurately, it is recommended that users consult previous editions as appropriate. In all cases, only the current base version of this publication is to be considered the official document.

Redline Processing Notes:

1. ~~Red Text~~ - Red strikethrough text denotes deletions.
2. Blue Text - Blue underlined text denotes modifications and additions.

Abstract: Identity assertion, role gathering, multilevel access control, assurance, and auditing are provided by the Biometric Open Protocol Standard (BOPS). The BOPS implementation includes software running on a client device, a trusted BOPS server, and an intrusion detection system. The BOPS implementation allows pluggable components to replace existing components' functionality, accepting integration into current operating environments in a short period of time. The BOPS implementation provides continuous protection to the resources and assurance of the placement and viability of adjudication and other key features. Accountability is the mechanism that proves a service-level guarantee of security. The BOPS implementation allows the systems to meet security needs by using the application programming interface. The BOPS implementation need not know whether the underlying system is a relational database management system or a search engine. The BOPS implementation functionality offers a "point-and-cut" mechanism to add the appropriate security to the production systems as well as to the systems in development. The architecture is language neutral, allowing Representational State Transfer (REST), JavaScript Object Notation (JSON), and Secure Sockets Layer (SSL) or Transport Layer Security (TLS) to provide the communication interface. The architecture is built on the servlet specification, open SSLs, Java, JSON, REST, and an open persistent store. All tools adhere to open standards, allowing maximum interoperability.

Keywords: admin console, application, BOPS admin, BOPS cluster, BOPS IDS, BOPS server, client device IDS, IDS cluster, IEEE 2410™, Jena Rules, liveness, original site admin, site admin, trusted adjudicated data, user, user device

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2017 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 20 October 2017. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-4321-0 STD22773
Print: ISBN 978-0-5044-4322-7 STDPD22773

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.” They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (“IEEE-SA”) Standards Board. IEEE (“the Institute”) develops its standards through a consensus development process, approved by the American National Standards Institute (“ANSI”), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Xplore at <http://ieeexplore.ieee.org/> or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at <http://standards.ieee.org>.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this IEEE standard was completed, the Biometrics Open Protocol Working Group had the following membership:

Scott Streit, *Chair*
Clayton Stewart, *Vice Chair*

Elizabeth Belousov

Bradley Boyer
Stephen Suffian

Mark Thompson

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Charles Barest
Sourav Dutta
Dan Friedman
Fernando Garcia-
Quismondo
Randall Groves

Marco Hernandez
Werner Hoelzl
Noriyuki Ikeuchi
Piotr Karocki
Stuart Kerry
Maximilian Riegel
Clayton Stewart

Scott Streit
Walter Struppler
Stephen Suffian
Mehmet Ulema
John Vergis
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 28 September 2017, it had the following membership:

Jean-Phillipe Faure, *Chair*
Gary Hoffman, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Chuck Adams
Masayuki Ariyoshi
Ted Burse
Stephen Dukes
Doug Edwards
J. Travis Griffith
Michael Janezic

Thomas Koshy
Joseph L. Koepfinger*
Kevin Lu
Daleep Mohla
Damir Novosel
Ronald C. Petersen
Annette D. Reilly

Robby Robson
Dorothy Stanley
Adrian Stephens
Mehmet Ulema
Phil Wennblom
Howard Wolfman
Yu Yuan

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 2410™-2017, IEEE Standard for Biometric Open Protocol.

Biometrics encryption, enhanced intrusion detection system (IDS) analytics, and advanced identification process distinguish the architecture design of IEEE Std 2410™-2017 from its predecessor IEEE Std 2410™-2015. The reinforced architecture of the Biometric Open Protocol Standard (BOPS) is well suited for implementation into enterprise systems for secure authentication via biometric modalities.

The biometric enrollment information, i.e., representation of a fingerprint, voice, facial features, is cryptographically protected into two parts and stored, respectively, on a client device and a remote BOPS server. Visual cryptography (VC) and the Euclidian distance matching algorithms make it nearly impossible to recover or reverse engineer one part of biometrics if another part is compromised.

With the increasing need to secure user's access to their footprints of personally identifiable information (PII) in the Internet (financial and health records) and enterprise assets, the BOPS server is designed to control communication with its clients via two-way secure socket layer/transport layer security (SSL/TLS) and to monitor authentication logs and patterns with IDS analytics. Figure 1 describes the authentication cycle for the BOPS framework.

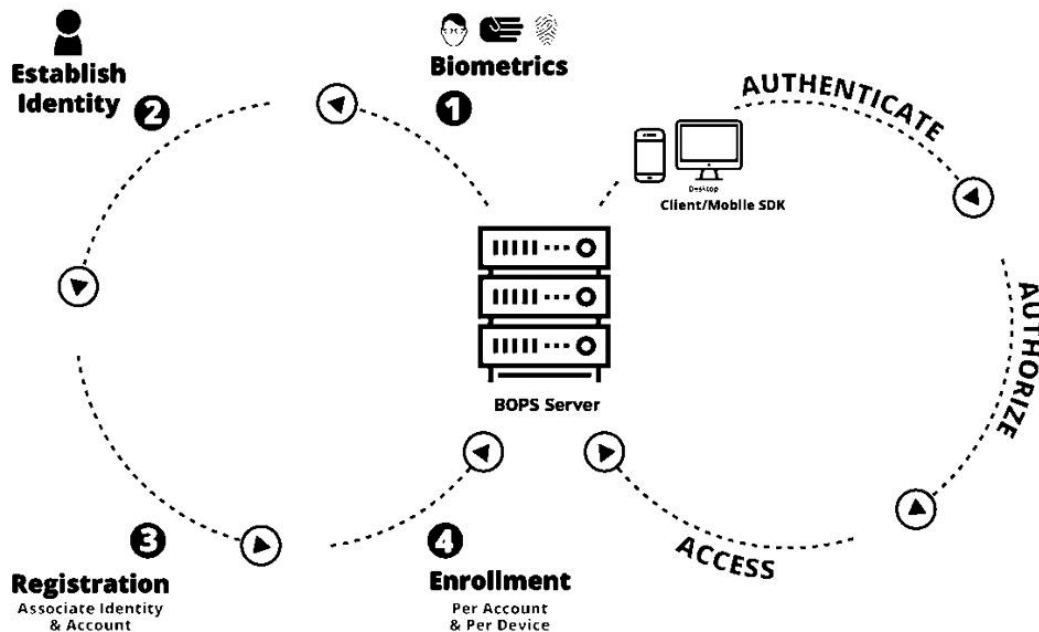


Figure 1—BOPS authentication cycle

Before users are being granted access via the BOPS platform, they shall authenticate their identity with an enterprise system that controls access to the resources and assets. If authentication is successful, the user is authorized to access the resource or asset (i.e., they are “granted access”). Otherwise, they are denied access. Identity is often established at some prior phase via registration of a username that identifies the user with an account in the system.

Biometrics have a long-held hope of replacing passwords by establishing a non-repudiated identity and providing authentication with convenience. Biometrics include a wide range of information taken from a person, e.g., fingerprints, face, voice, and iris pattern, and his/her behavioral properties, e.g., gait, date, time, and location. Recent increases in the processing power and sensor technologies allow digital signal processing (DSP) algorithms to run in the time needed for a real-time authentication (1–5 s similar to username and password login processing). Unlike passwords, biometrics cannot be script-injected; however, biometric data is considered highly sensitive due to its personal nature and unique association with users. Secure storage, transport, and processing of biometric data is paramount in the design and implementation of the BOPS system.

Where IEEE Std 2410-2015 signified a new approach in Identity and Access Management that enabled a password replacement and multi-factor authentication, the IEEE Std 2410-2017 brings a new level of consumer privacy assurance by keeping biometric data encrypted at rest and transit.

Contents

1. Overview	11
1.1 Scope	11
1.2 Purpose	11
1.3 Intended audience	11
2. Normative references	11
3. Definitions, acronyms, and abbreviations	12
3.1 Definitions	12
3.2 Acronyms and abbreviations	12
4. Conformance	13
5. Security considerations	14
5.1 Background	14
5.2 Identity assertion	14
5.3 Role gathering	14
5.4 Access control	15
5.5 Auditing and assurance	15
6. BOPS interoperability	16
7. BOPS overview, application, registration, and prevention of replay	17
7.1 Overview	17
7.2 Application	20
7.3 Security architecture	21
7.4 System overview	22
7.5 Solution architecture	24
7.6 Biometric engines and their scopes	26
7.7 Genesis	26
7.8 Enrollment	27
7.9 Biometric matching with visual cryptography	27
7.10 Homomorphic encryption	31
7.11 Defaults	34
7.12 Authentication Requirement	35
7.13 Enrollment requirement	35
7.14 Registration	35
7.15 Prevention of replay	37
8. BOPS infrastructure	39
8.1 BOPS DNS	39
8.2 BOPS TrustStore	39
8.3 BOPS KeyStore	39
8.4 Key negotiation protocol	40
8.5 Enrollment elements	40
8.6 Inside the BOPS infrastructure	40
8.7 Client roles	40
8.8 Message encoding and decoding	40
8.9 Data privacy	41
8.10 Genesis logical flow	42
8.11 Certificate distribution	45
8.12 Certificate management policy	48

9. BOPS API overview	49
9.1 Format	49
9.2 Identity assertion API	50
10. API	51
10.1 Enterprise concepts.....	51
10.2 Format of API cells.....	51
10.3 The start of the biometric workflow	51
10.4 Authentication overview.....	52
10.5 API—genesis.....	53
10.6 API enrollment	55
10.7 Data Structure.....	57
10.8 API—QROpportunity	58
10.9 Client side authentication	61
10.10 Server side authentication.....	62
10.11 Biometric engines configuration.....	67
10.12 Application settings.....	68
10.13 Business integration	69
10.14 Role gathering API	70
10.15 Access control API	72
10.16 Auditing.....	73
10.17 Administration.....	73
10.18 Reporting.....	74
10.19 Admin statistics API	74
11. Server-side intrusion detection system	75
11.1 API list blacklist.....	75
11.2 API—incident.....	75
12. Client device requirements	76
13. Privacy considerations	76
13.1 Background	76
13.2 BOPS data privacy reference.....	77
13.3 BOPS governance and compliance.....	77
13.4 BOPS PII	79
13.5 BOPS privacy specific safeguards	79
13.6 BOPS and privacy controls.....	81
Annex A (informative) Glossary	84
Annex B (informative) Bibliography	85

IEEE Standard for Biometric Open Protocol

~~IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.~~

~~This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.~~

1. Overview

1.1 Scope

The Biometric Open Protocol Standard (BOPS) provides identity assertion, role gathering, multilevel access control, assurance, and auditing. The BOPS implementation includes software running on a client ~~device~~ (e.g., ~~smartphone~~ [web](#) or ~~mobile-device~~), a trusted BOPS server, and an intrusion detection system (IDS). The BOPS implementation allows pluggable components to replace existing components' functionality, accepting integration into the current operating environments in a short period of time. The BOPS implementation adheres to the principle of continuous protection in adjudicating access to resources. Accountability is the mechanism that proves a service-level guarantee of security. The BOPS implementation allows the systems to meet security needs by using the application-programming interface (API). The BOPS implementation need not know whether the underlying system is a relational database management system (RDBMS) or a search engine. The BOPS implementation functionality offers a "point-and-cut" mechanism to add the appropriate security to the production systems as well as to the systems in development.

BOPS includes authentication with splitting of the initial biometric vector (IBV), sometimes called the initial template, optionally into one or two pieces. Irrespective of the number of pieces, the IBV is encrypted in a keyless fashion and the subsequent biometric match optionally occurs on the client or on the server, as denoted by an administration parameter.

1.2 Purpose

This standard provides a means to gather biometric ~~agnostic~~, data using a multi-level security protocol for authentication, identification, and access control and assurance.

1.3 Intended audience

The intended audience of this document includes security evaluators, system underwriters, developers, and systems engineers. The BOPS is subject to changes and updates.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

~~The~~ DoD 5200.28-STD, Trusted Computer System Evaluation Criteria, 1985 ~~(Mitre 1984)~~.

ISO 8601, Data elements and interchange formats—information interchange—representation of dates and times.¹

¹ISO publications are available from the ISO Central Secretariat (<http://www.iso.org/>). ISO publications are also available in the United States from the American National Standards Institute (<http://www.ansi.org/>).

3. Definitions, acronyms, and abbreviations

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.²

3.1 Definitions

account: A user account that was validated (against an external system or by an email validation mechanism). It ~~can have~~ is associated with one or ~~multiple mobile~~ more client devices. The enrollment process ends by creating a client certificate for the device that shall be used for subsequent calls to authenticate against the ~~HoyesID~~ BOPS platform.

Bell-LaPadula: The multilevel model that was proposed by Bell and LaPadula for enforcing access control in government and military applications. A subject ~~can only~~ access accesses objects at certain levels determined by their security level.

JUnit: A testing framework for Java programming language.

RESTful: Refers to Representational State Transfer (REST), which is a software architecture style.

SHA512: Secure hash algorithm one, which was designed by the US. National Security Agency (NSA).

3.2 Acronyms and abbreviations

1:M one-to-many

4F Four fingers (a proprietary biometric technology)

5-tuple five tuple data parameters

²IEEE Standards Dictionary Online ~~subscription~~ is available at:
~~http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html~~ <http://ieeexplore.ieee.org/xpls/dictionary.jsp>.

admin	administrator
~~AOP~~	~~aspect-oriented programming~~
API	application programming interface
<u>BLOB</u>	<u>binary large object</u>
~~app~~	~~a mobile client application~~
BOPS	biometric open protocol standard
<u>CA</u>	<u>certificate authority</u>
<u>CBV</u>	<u>current biometric vector</u>
<u>CN</u>	<u>common name</u>
CPU	central processing unit
<u>CSR</u>	<u>certificate sign request</u>
<u>DDoS</u>	<u>distributed denial of service</u>
<u>DN</u>	<u>distinguished name</u>
<u>DNS</u>	<u>domain name server</u>
<u>IBV</u>	<u>initial biometric vector</u>
~~CSRF~~	~~cross-site request forgery~~
ID	identifier
IDS	intrusion detection system
IP	internet protocol

JSON	JavaScript object notation
LDAP	Lightweight Directory Access Protocol
M:M	many-to-many
MAC	mandatory access control
MCA	mobile client application
MILS	Multiple Independent Levels of Security/Safety
NSA	National Security Agency (US)
OTP	one time password
PII	personal identifiable information
QR code	Quick Response code
RDBMS	relational database management system
REST	representational state transfer
ROTP	rapid one time pad
SDK	software development kit
SOLR	Searching on Lucene with Replication
SSL	secure socket layer
TCSEC	trusted computer system evaluation criteria
TLS	transport layer security
TTL	time-to-live
~~XNTP~~	~~extended network time protocol~~

URL	uniform resource identifier
VC	visual cryptography
VCS	visual cryptography scheme
VPN	virtual private network
XOR	“exclusive OR” (a binary operation)

4. Conformance

The BOPS comprises the rules governing secure communication between a variety of client devices and the trusted server. This standard is based on the tested computer-based implementation of the Trusted Computer System Evaluation Criteria (TCSEC).³

BOPS conforms to the TCSEC, which is the US. Department of Defense standard that sets basic requirements for assessing the effectiveness of computer security controls built into a computer system. TCSEC was created by the National Computer Security Center, an arm of the National Security Agency (NSA) and is also frequently referred [to](#) as “Orange Book, section B1.” BOPS also conforms to the Director of the Central Intelligence Directive 6/3 protection level 3, level 4, and level 5 (PL3, PL4, and PL5), and to the standards of the Multiple Independent Levels of Security/Safety (MILS) architecture.

5. Security considerations

5.1 Background

Security considerations include the security policies in place and unambiguously defined levels of security. One of the BOPS’s main functions is to provide authentication instead of authorization in a way that the server does not retain the client information but rather recognizes one client from another. The key components of security considerations include identity assertion, role gathering, access control, auditing, and assurance.

³[Information on references can be found in Clause 2.](#)

BOPS largely considers the server side component to an end-to-end biometric solution. The initial solution was recorded as IEEE Std 2410-2015 and described all the components necessary for the server side of end-to-end biometric security. The standard lists requirements for the client component to comply with the Server Side Solution. The standard describes a set of new technologies including but not limited to liveness. "Liveness" is a term that means insuring the biometric is from a live source, such as a human being, and not from an imitation, such as a wax replica, a picture, or a video.

This standard further enhances the security model to mitigate recently uncovered vulnerabilities. Security itself is a constant game of "cops" and "robbers". The more sophisticated the adversaries ("robbers") become, the more powerful the security model or "cop" required. This standard raises the bar in all areas of biometric security with a diligent focus on authentication and privacy. Additionally, we tackle the difficult problem of biometric identification with the maintenance of privacy.

Users no longer have to remember the password, or worse yet, have their passwords end up in the wrong hands. Biometrics are part of us, unique to us, and excellent for leveraging in a secure solution.

Using multiple rounds of penetration tests, the IEEE vetting process, and industry and client critiques, BOPS has been validated. This mature and secure framework provides an end-to-end solution without the compromise of privacy.

5.2 Identity assertion

The BOPS implementation helps provide continuous protection to resources and assurance of the placement and viability of adjudication and other key features. Accountability is the mechanism that ~~can help~~ helps provide a service-level guarantee of security.

The BOPS implementation identity assertion helps confirm that named users are who they claim to be. The identity assertion implies reliance on human biometrics; however, the BOPS is an interoperable standard and ~~can~~ [is able to](#) incorporate any identity asserter, or a number of different asserters, associated with the same named user. The application of the IDS provides active monitoring to help prevent spoofing of the credentials set and to blacklist a subject or device that makes malicious attempts.

5.3 Role gathering

Role gathering is focused on the data confidentiality and privileged access that is based on the rules enforced by a known system. To determine whether a specific access mode is allowed, the privilege of a role is compared to the classification of the group to determine if the subject is authorized for a confidential access. The object's structure is defined by the access control. Role gathering occurs on the system level or through the client/server call. The BOPS implementation server stores role gathering information to associate a unique user with a unique device.

5.4 Access control

5.4.1 General

Access control asks whether a given subject (person, device, or program) ~~may~~ [has the ability to](#) read, write, execute, or delete a given object. The community further divides access control into discretionary access control and a more granular form of access control called mandatory access control (MAC).

5.4.2 Discretionary access control

The BOPS implementation supports access control between the named users and the named objects (e.g., files and programs). The adjudication mechanism is role based and allows users and administrators to specify and control sharing of those objects by named individuals and defined groups of individuals. The discretionary access control mechanism provides that objects are protected from unauthorized access.

Discretionary access control provides protection at the group or individual level across a single or group of objects. The granularity ranges from individual to group.

5.4.3 Mandatory access control

The BOPS implementation shall enforce a mandatory access control policy over all subjects and storage objects under its control (e.g., processes, files, segments, devices). These subjects and objects are assigned sensitivity labels, which are a combination of hierarchical classification levels and nonhierarchical categories. The labels are used in the adjudication as the basis for mandatory access control decisions. The client software shall maintain labels or have the BOPS server maintain the data in order to force adherence to labeling of the subject and objects. The BOPS implementation server maintains a trusted store as a component of BOPS.

The following requirements hold all accesses between subjects and objects controlled by the BOPS:

- A subject ~~can~~ has the ability to read an object only if the hierarchical classification in the subject's security level is greater than or equal to the hierarchical classification in the object's security level.
- The nonhierarchical categories in the subject's security level include all the nonhierarchical categories in the object's security level.

- A subject ~~can~~ has the ability to write an object only if the hierarchical classification in the subject's security level is less than or equal to the hierarchical classification in the object's security level and all the nonhierarchical categories in the subject's security level are included in the nonhierarchical categories in the object's security level.
- Identification and authentication data ~~should~~ shall be used by the BOPS to authenticate the user's identity and ~~to maximize that~~ the security level and authorization of subjects external to the BOPS that ~~may be~~ are created to act on behalf of the individual user are dominated by the clearance and authorization of that user.

5.5 Auditing and assurance

5.5.1 Background

The worst possible case is when a system is compromised and the compromise goes undetected. To help prevent this case, the aforementioned specifications require auditing and proof of the security model, which is called assurance.

5.5.2 Assurance

The current IEEE 2410 specifications propose to conduct assurance by running the sets of testing and test programs with RESTful API (application programming interface), and aggregate output data (test results) for further analysis.

Prior to the beginning, there are constructed test plans that shall reflect backward compatibility, systems update compatibility, and new installation scenarios.

Planning test cases for assurance includes (but is not limited) to: functional testing, business logic validation, test for data integrity, test for lack of non-repudiation, test for trust relationship, test for process timing, and test for bypassing workflows.

5.5.3 Auditing

The BOPS implementation supports all auditing requests at the subject/object level or at the group level. The BOPS implementation uses aspect-oriented programming to ~~maximize the likelihood~~ [guarantee](#) that all calls are safely written to an audit trail. An interface of RESTful web services and JavaScript object notation (JSON) provides a mechanism to read the audit trail. Auditing ~~may~~ [will](#) [either](#) occur at the subject per action, the object per action, or the group per action. For example, a group of users called “accounting” ~~may~~ [will](#) audit all writes to [the](#) general ledger, or the chief financial officer ~~may~~ [will](#) have audits for reads of the income statement.

5.5.4 System integrity

JUnit tests exist for all boundary conditions of the BOPS. The suite of tests includes testing all boundary components and conditions ([such as a backward compatibility](#)) of the system. [The BOPS boundary components are:](#)

- [BOPS instance/BOPS security group](#)
- [Identity for user\(s\)](#)
- [Role gathering for users](#)
- [Object access control](#)
- [Node failures, singular](#)
- [Node failures, multiple](#)
- [Account lock and unlock](#)

[In addition to that we recommend conducting regular security testing and assessments that adhere to the industry or company security standards.](#)

6. BOPS interoperability

The BOPS implementation allows the systems to meet security needs by using the API. The BOPS implementation does not need to know whether the underlying system is ~~an~~ a RDBMS or a search engine. The BOPS implementation functionality offers a “point-and-cut” mechanism to add the appropriate security to the production systems as well as to the systems in development. The architecture is language neutral, allowing REST, JSON, and SSL to provide the communication interface. The architecture is built on the servlet specification, openssl, Java, JSON, REST, and a persistent store. All tools adhere to open standards, allowing maximum interoperability as shown in Figure 2.

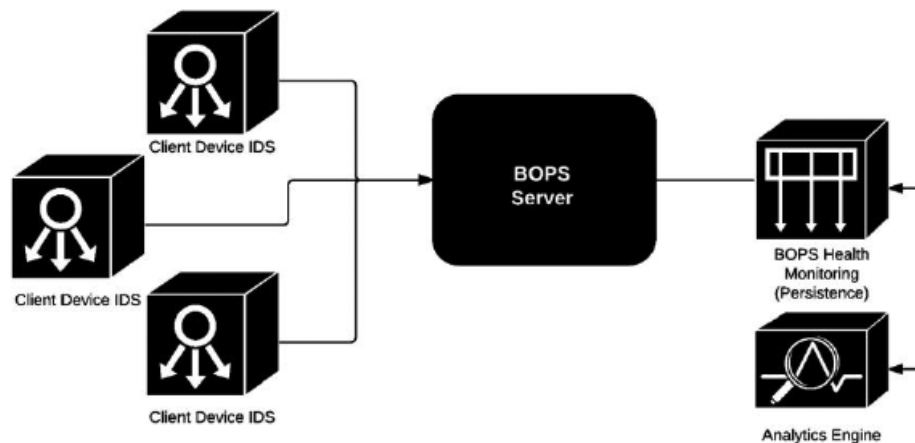


Figure 2—BOPS system diagram

7. BOPS overview, application, registration, and prevention of replay

7.1 Overview

The BOPS implementation provides identity assertion, role gathering, multilevel access control, auditing, and assurance. The BOPS implementation includes software running on a client device (~~e.g., smartphone or mobile device~~), a trusted BOPS server, and the ~~intrusion detection system (IDS)~~. The BOPS implementation allows pluggable components to replace existing components' functionality, accepting integration into current operating environments in a short period of time.

The client/device application loads a one-time, two-way SSL/TLS key for the initial communication to the server. This key is replaced, in function, by the subject's two-way SSL/TLS key during the Identity Genesis phase. Identity Genesis is an initial step that fuses a set of biometrics with a given subject.

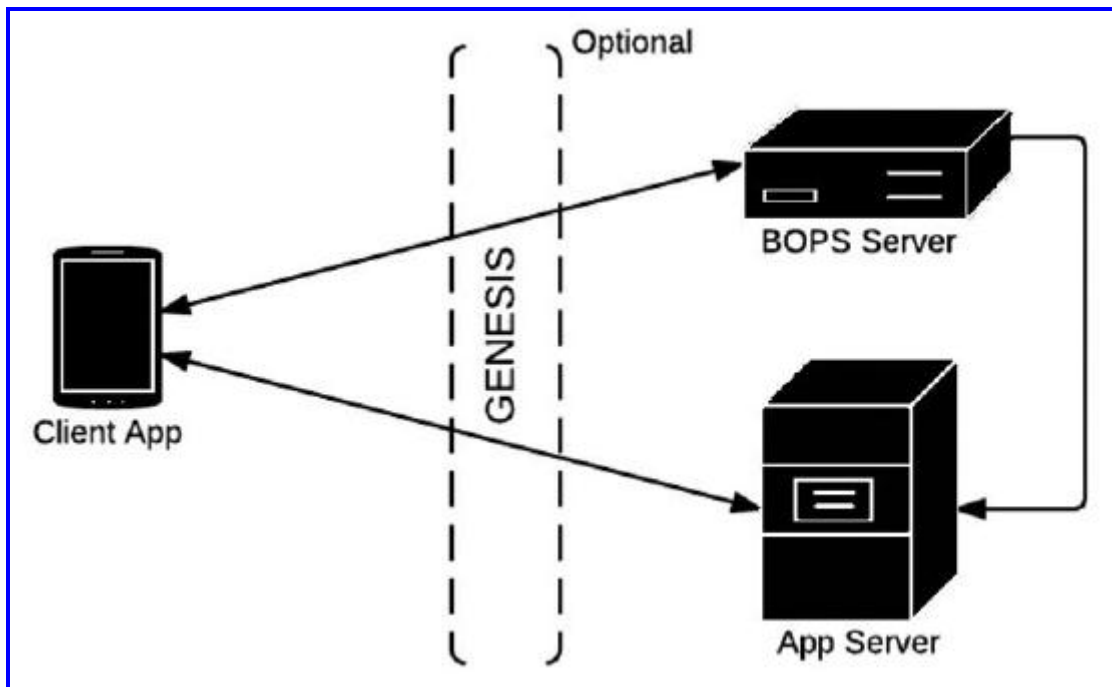
The client/server application's interaction with the BOPS could be described as a three-step process with two possible variations after the first step. For the purposes of this standard, the BOPS client/server application architecture is presented in three components: client application, the BOPS security server, and the server-side application (called "App Server" in the following figures).

As shown in ~~Figure 2,~~ Figure 3 ~~through Figure 4, Figure 5, and Figure 6~~ [7](#), the most important feature of the BOPS client/server application connection is that the server-side application is not running through the BOPS server and the SSL/TLS connection terminates at the application server. The BOPS implementation deployment does not require the application to trust the BOPS system with the unencrypted content.

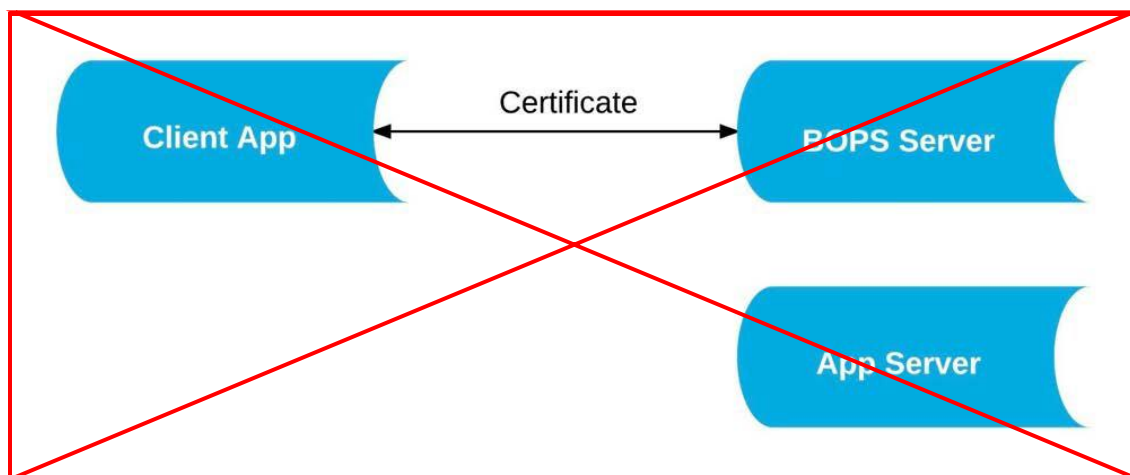
The most notable difference in the BOPS is that the implementation of Genesis (see 7.6) is optionally enforced by the third-party system business logic, as depicted in Figure 3.

In the first variation, during optional Genesis (or Enrollment), the client makes a call to the BOPS server, authenticates the user and client-side application software, and then receives, allocated by the BOPS server, a certificate that is specific to the client identity of a specific application. It should be noted that the client contains a pre-loaded SSL/TLS certificate.

During the next step (Figure ~~3~~ [4](#)), the client device/application calls the BOPS server directly. The client application ~~server directly;~~ creates a certificate request, and the BOPS server signs the certificate during the Register Account call. The IBV, in half form, is sent to the server. Other values, which mitigate replay, are sent to the server and the client ~~is not on this path. The SSL/TLS connection between the client and server parts of the application starts and terminates at these points. All content exchange is not visible outside of the application to the BOPS server or others not trusted within this application entity.~~

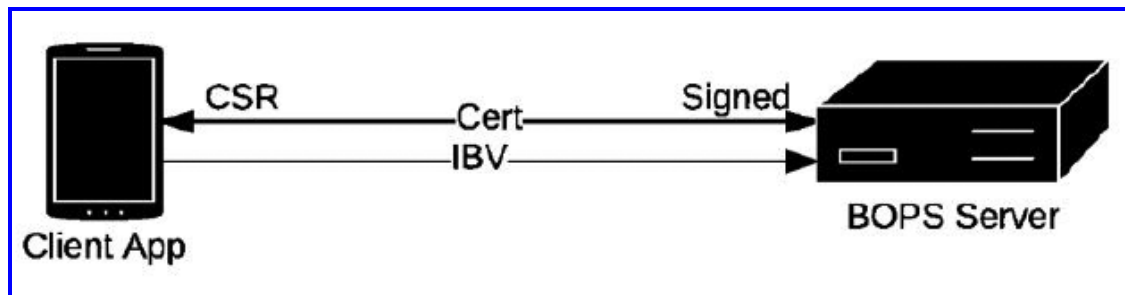


(New)

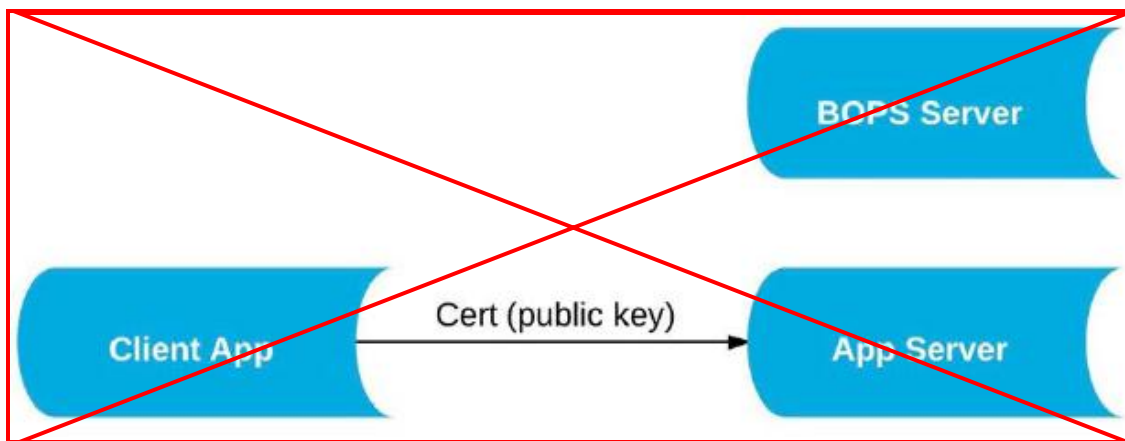


(Deleted)

Figure 3—Client/server application architecture: Genesis



(New)



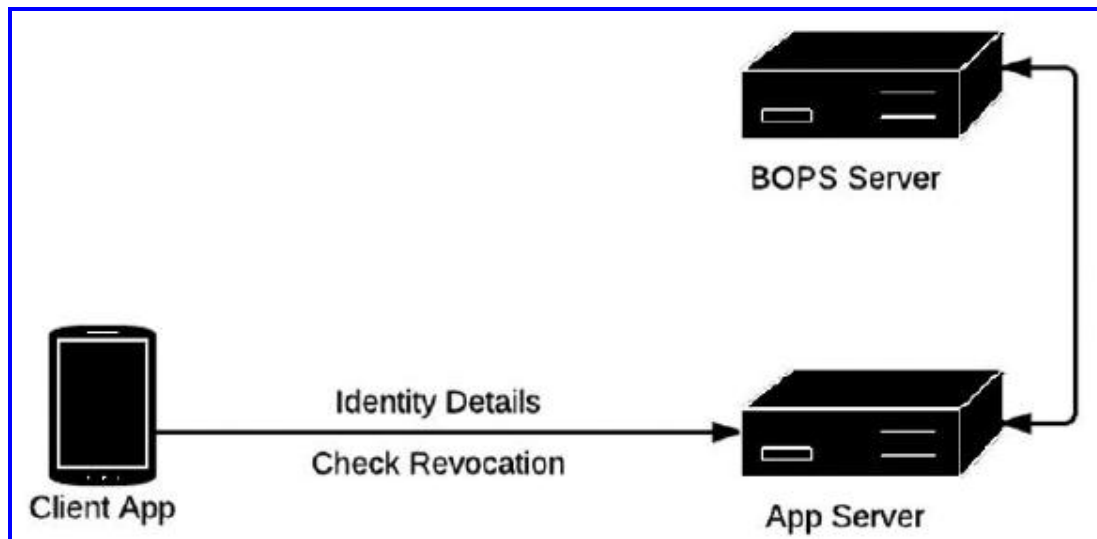
(Deleted)

**Figure 4 – Client/server application architecture: Enrollment ~~Client calls~~
server application**

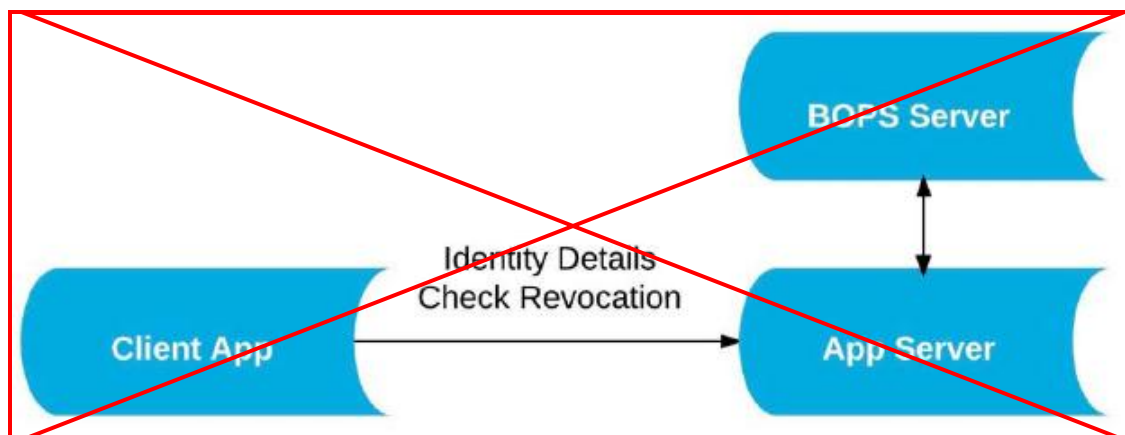
During the client session (Figure 4 5), the application server calls the BOPS server to get identification details and confirms the certificate has not been revoked already.

In the second variation (Figure 5 6), the first step is the same: Genesis. After the first step is completed, the BOPS server contacts the application server component to notify that a new client has been registered and allocated.

The flow of this second variation (Figure 5 6) differs from the first ~~variation's~~ variation flow (Figure 3 4) in two ways: the identity details are different, and the revocation check is procured in the client session (Figure 6 7).

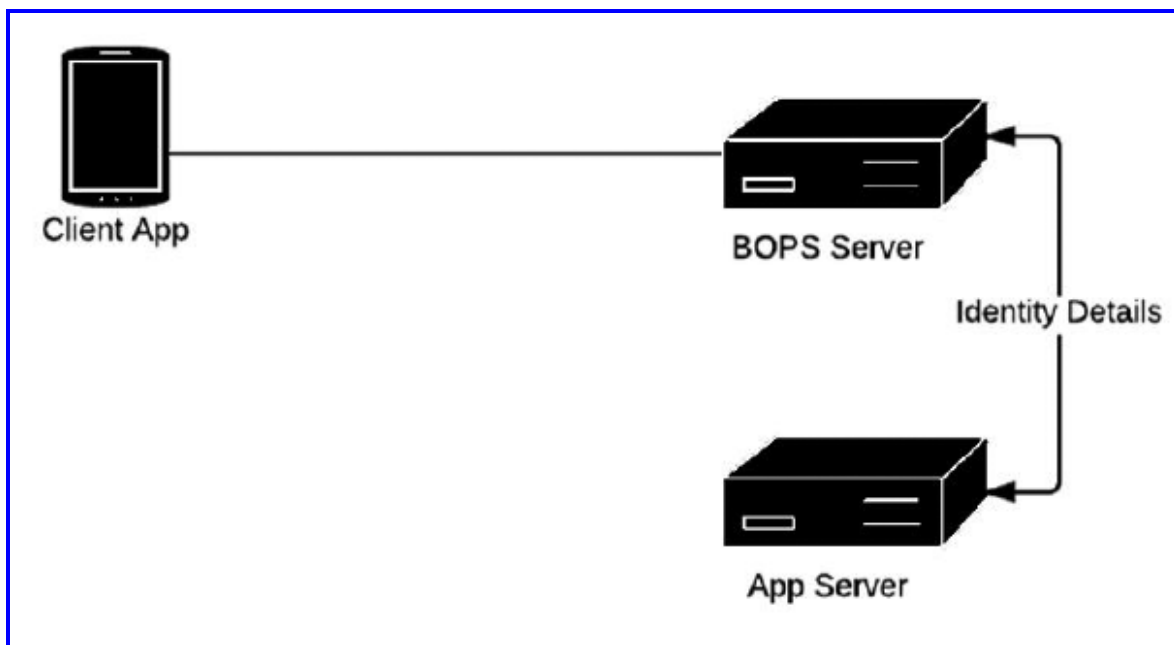


(New)

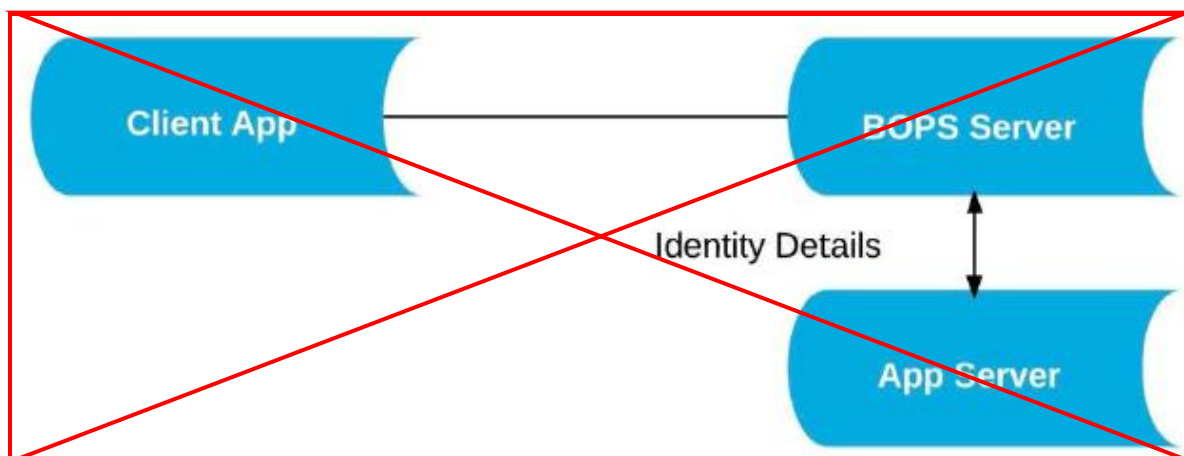


(Deleted)

Figure 5—Client/server application architecture: Client session

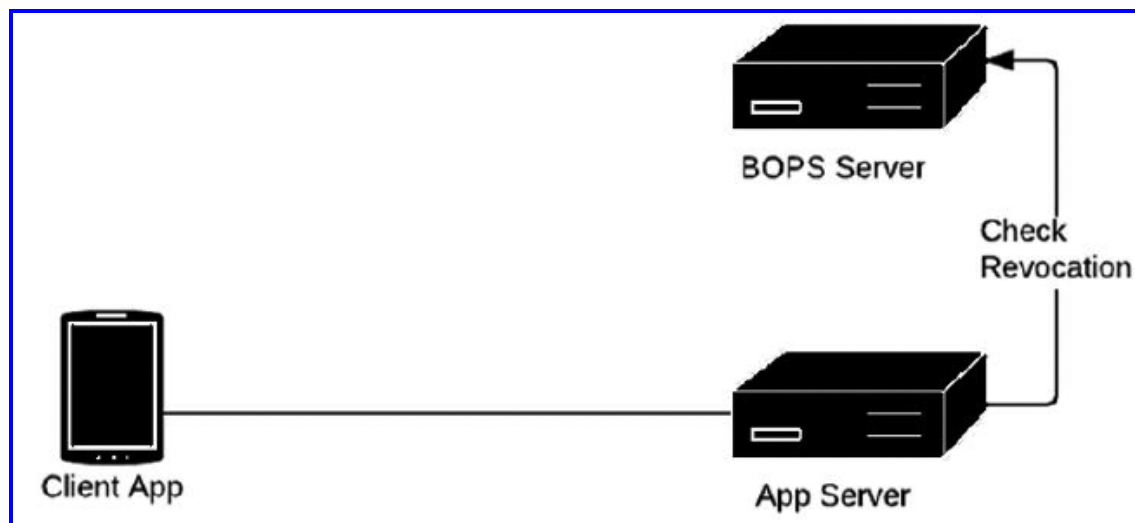


(New)

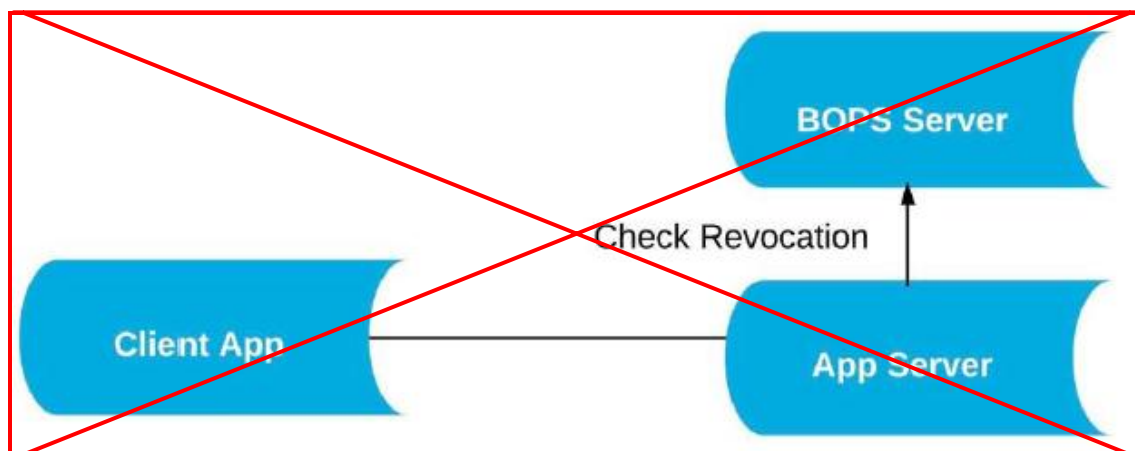


(Deleted)

Figure 6— Client/server application architecture: Variation of client-calls-server application



(New)



(Deleted)

Figure 7— Client/server application architecture: Variation of client session

At the third step, when the client calls the application directly, the application server calls the BOPS server to check if the certificate has not been revoked yet.

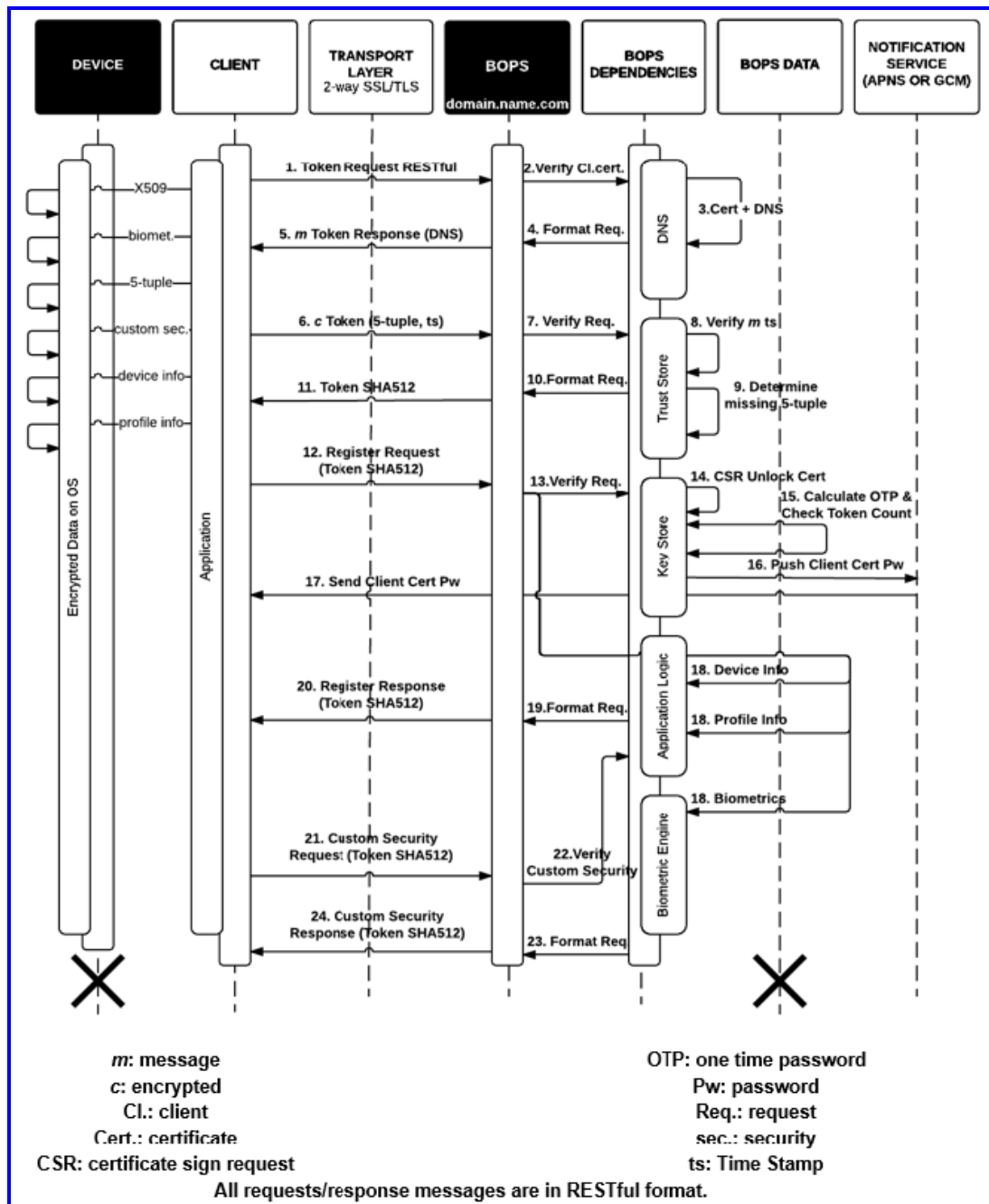
7.2 Application

The entire BOPS suite could be used through the access control or simply be added to the identity assertion of an already existing framework. The BOPS implementation enables trusted processing by performing the minimum actions in the production environment and, in most cases, does not require the change of any application software. [The entire data confidentiality flow of the BOPS implementation is visualized in Figure 8.](#)

Two-way SSL/TLS, which is built on top of one-way SSL/TLS, provides communication starting at the client. The initial, or Genesis, communication establishes the origin of the client's identity and passes the BOPS-compliant two-way certificate that the client uses for subsequent communication in conjunction with the session-oriented identity assertion. It is important to note that the client application shall have a pre-loaded two-way SSL/TLS key that allows subsequent identityGenesis.

The BOPS implementation compliant server receives one-way SSL/TLS communication with two-way SSL/TLS identity. Communication is conducted via both one-way SSL/TLS and two-way SSL/TLS. The server uses a data store to take trusted identity and gather the roles for processing on behalf of the identity. Auditing ~~maximizes~~ [increases](#) the appropriate artifacts for continued verification and validation of the trusted access. The assurance occurs through the simplification and documentation of the multilevel access control mechanism. The BOPS implementation requires an administration console (hereafter "admin console"), which is available after the registration process (see 7.3 [14](#) for BOPS registration), ~~which~~ [that](#) allows dynamic modification of users, groups, and roles.

BOPS shall be implemented with an active IDS [on the server](#) that ~~provides~~ [helps to provide](#) prevention of any form of brute-forcing or denial-of-service (distributed or single denial-of-service) attacks. The standard contains a custom rule that identifies and tracks the attempts to forge two-way SSL/TLS certificate impersonation, a session replay, forged packets, and a variety of other attempts to circumvent the BOPS server. [See The network communication between the BOPS clusters in the enterprise system and the clients are configured through the firewall, as shown in](#) Figure [9](#) ~~7~~.

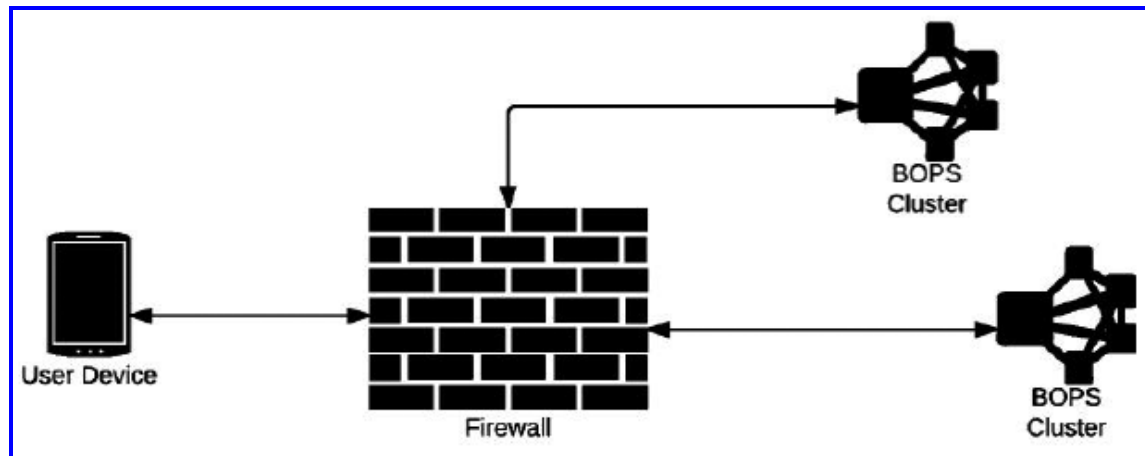


(New)

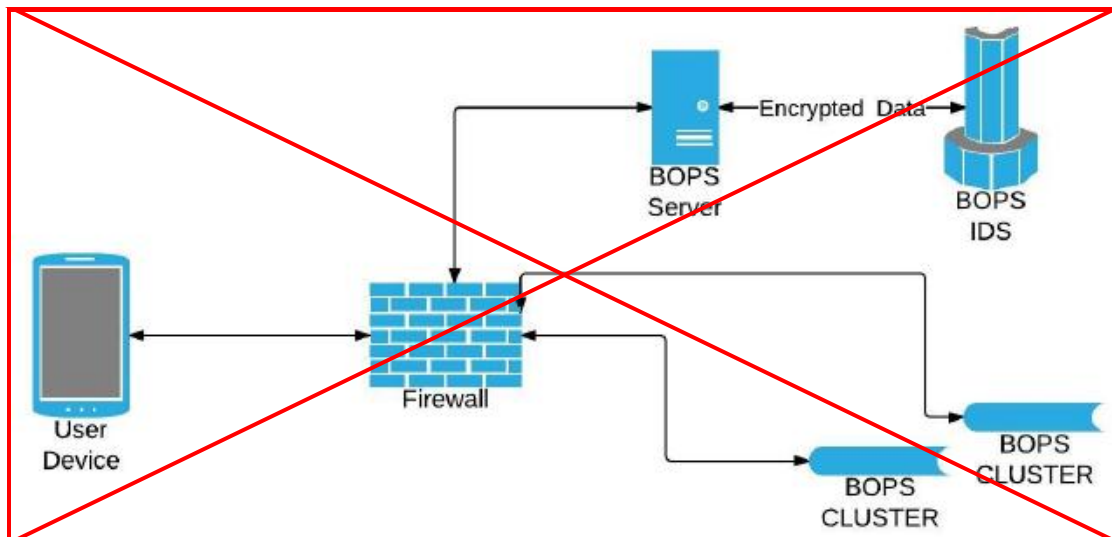
Figure 8—Data confidentiality flow during enrollment

7.3 Security architecture

There are several critical components of BOPS security architecture that could be divided into several architectural components as illustrated in Figure 10.



(New)



(Deleted)

Figure 9— Client/server application architecture: Variation of client session

7.3.1 BOPS cluster

BOPS security group is responsible for running the BOPS instances on the virtual private networks (VPNs). The core attributes of a Certificate Authority entity, the BOPS KeyStore and the BOPS TrustStore are located on the BOPS instances. The BOPS instances also contain data about the domain name server (DNS), OTP (one-time password) library, notification services keys, business adapters, and BOPS configuration properties.

7.3.2 Load balancer cluster

The load balancer cluster is responsible for reliability of BOPS services, distributed workload, and the high availability of BOPS. A configured BOPS load balancer helps to maximize throughput, helps to minimize response time, and helps to avoid overload of any single resource in the BOPS architecture.

7.3.3 Persistence cluster

Data security group is responsible for auto-scaling of BOPS data clusters. As the biometric engines essentially deal with the large data objects, handling large data sets requires the big data store in BOPS. These types of data stores are normally called NoSQL because of their non-relational nature. In the non-relational paradigm, data sets are called shards, a horizontal partition of data. Big data solutions improve performance by reading from shards concurrently and merging the results. The data security architecture serves BOPS philosophy of helping to prevent centralized storage of sensitive data in one place.

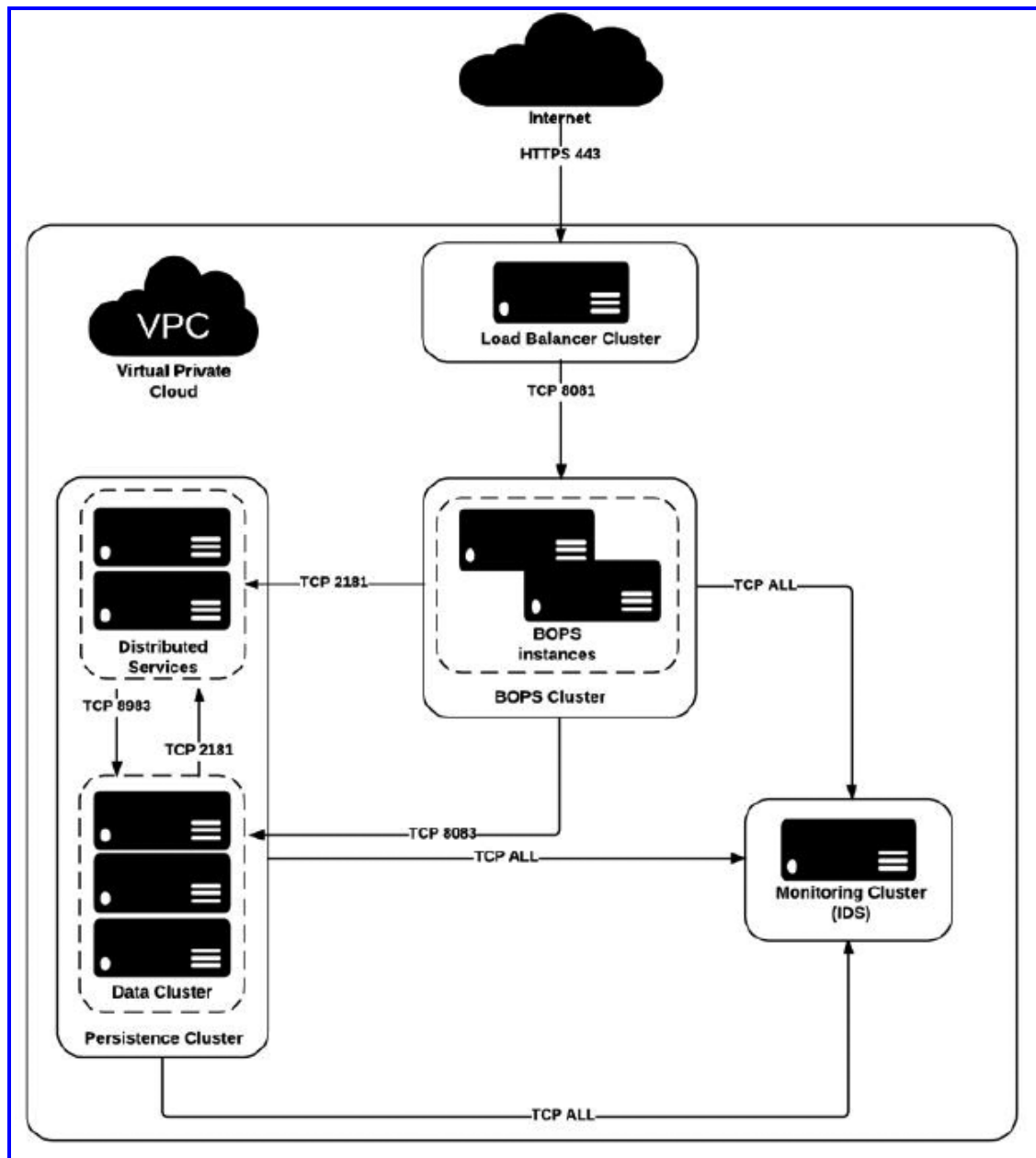
7.3.4 Monitoring cluster

The monitoring cluster uses machine learning and other data analytics techniques in finding access patterns of ill intent. This call automatically triggers actions as an IDS or shall produce reports and other access to support security administrators.

7.4 System overview

BOPS uses visual cryptography (VC) to encrypt the IBV. This technique offers the advance of fast recomposing of the initial vector using “exclusive or” (XOR) operation on whichever device does the biometric match.

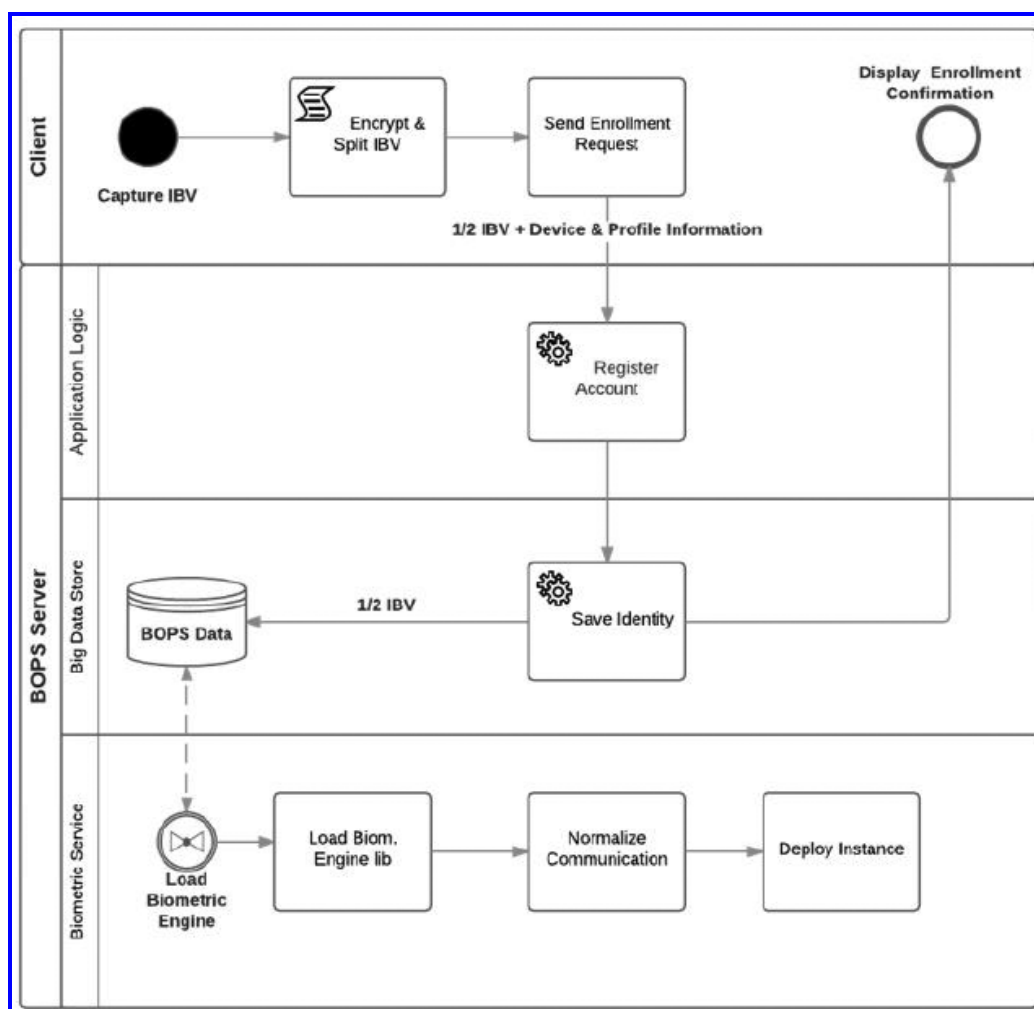
We choose to use Moni Naor’s and Adi Shamir’s technique, which provides a secret sharing scheme. Basically, a vector is broken in N shares and recomposing the initial vector shall require all N share parts. In our case, because the parties implied are the BOPS server and mobile client application (MCA), the enrolled vector shall be broken in two shared parts, one being stored in the BOPS repository and the other one on the client during the enrollment process. The share part on the client side shall be stored in a secure element, provided that the application has access to that secure element. Otherwise, it is stored in the user space.



(New)

Figure 10—BOPS security architecture diagram

Consequently, during the biometric authentication action, a new acquired vector and both shares of the enrolled vector should be available in the same place (the client application or the BOPS server). Using enrolled vector shares, the initial vector shall be recomposed in memory, and the authentication shall be performed against it. Some of the biometric vector must be kept in memory (unfortunately creating an attack window), as there are no currently available techniques to perform biometric authentication on completely encrypted data. The subsequent infrastructure and flows are designed based on Figure 8 and Figure 10. Figure 11 shows the encrypted authentication data stored between BOPS server and client application after a successful enrollment.



(New)

Figure 11 – Enrollment process simplified

The next subclauses shall describe the details related to this aspect and with the other mechanism already in place, such as a) issuing a client certificate, b) protecting the client certificate, and c) protecting sensitive data stored on the client.

7.5 Solution architecture

A server side biometric engine integration was introduced in this version of IEEE Std 2410-2017. In the context of BOPS design architecture, it implements the following:

- Mechanism of storing encrypted biometric data associated with an account or a device
- Mechanism of storing information about all biometric data changes
- Generic mechanism of loading and using the biometric libraries (e.g., FACE, 4F, IRIS)
- Providing a new version of the API operations to support the new flows (enrollment and authentication)

A client application is responsible for the following:

- The IBV acquisition
- Performing the encrypted split operation during the enrollment process and distributing this process for a lower CPU (central processing unit) load on the server side
- Performing enrollment request (registration) with the BOPS server
- Performing a match operation when the biometry validation method from the authentication flow is configured to take place on the client side

The BOPS server is responsible for the following:

- Saving identity data together with the shared vector in the BOPS big data store during the enrollment process
- Managing the authentication flow on the server side or client side based on configuration

- Integrating the biometric service communication component

The biometric service is responsible for the following:

- A dynamic loading of the biometric engines library
- Supporting of the biometric engines versioning
- Normalizing communication between the BOPS server and the biometric engines
- Offering the different deployment scenarios (on the machines where the BOPS instances shall reside as separate clouds that scale by themselves)

The biometric engines perform the following:

- The unmanaged biometric libraries shall be a subject of the contract (an interface) that is defined and implemented by each library for plugging into the BOPS system
- Shall offer a Load method for loading a biometric engine if needed
- Shall offer UnloadLoad method for unloading an engine and freeing all resources used by it (e.g., memory, temporary files) if needed
- Shall offer GetStatus method to give status information (e.g., INIT FAILED, OK, ERROR, OUT_OF_MEMORY) if needed
- Shall offer Split method used to encrypt acquired vector during enrollment
- Shall offer Match method used to authenticate a vector based on the shared parts of the initial vector
- Shall offer Activate/Register method if needed
- Shall offer engine descriptions for the following:
 - Biometric type identifier

- Name and description
- Engine version
- Biometric format

Based on this information, the system loads and registers a specific biometric engine.

7.6 Biometric engines and their scopes

As explained above, the new BOPS provides a mechanism for pluggable biometric engines. This means the system shall be configurable (biometric libraries location) and it shall automatically load the available libraries and the register into the system. Each biometric library should describe itself by providing a constant string (biometric type), its version, and its name and description. The pair (BiometricType, BiometricVersion) identifies a unique biometric engine.

As a result, a list of biometric engines shall be available at the system level, for instance:

- Four fingers (4F)-engine
- FACE-engine
- IRIS-engine
- VOICE-engine

7.7 Genesis

A key component of the end-to-end solution is a decoupling of Genesis and Enrollment. Genesis is a server- side component for how we determine the identity of a subject without concern for the biometric, certificate, or any other automated processing.

The BOPS solution is open and it is capable of enabling any customization for a Genesis flow. Some examples of Genesis could include a username and password access to Active Directory, a validating email or text message, or an officer of an organization physically verifying identity.

Pre-registration of a user account that occurs in batch is based on business requirements. Ultimately, Genesis forms a full dependency on risk management and shall determine downstream processing.

Following Genesis, a user enrolls his or her biometric(s). The enrollment process includes a unique client certificate issued for the enrolled device. Additionally, a one-time password (seed) is established between the client and server. An additional seed value is incorporated to help prevent replay attack.

A user is capable of having many devices, and a device is capable of having many users. A device is capable of having many biometrics. This many-to-many relationship occurs through the separation of Genesis and Enrollment. The identified subject through Genesis is, therefore, able to enroll many times with many biometrics.

The enrollment process uses a two-way SSL/TLS certificate, which is server generated. This generation occurs after Genesis, guaranteeing that the certificate is for the well-defined subject.

The BOPS platform has provisions that give flexibility for different security levels. The highest level of Genesis shall be the user validated in front of an officer. The lowest level shall be a username and password in conjunction with a validating email. At the end of the day, the levels of Genesis and the verification process is a business decision frequently unique to each and every organization.

Subsequent processing changes based on the Genesis level. For example, a system shall allow a \$100 000 transfer for a high level of Genesis, but only a \$100 transfer for a lower level of Genesis.

Figure 12 illustrates possible requirements and examples for different levels of Genesis. Security level grows as we add more requirements to the verification process. First and second levels can be swapped based on the organizational considerations. If the goal is to verify and give WiFi access to business visitors, then verification sent via a mobile phone can be considered as a minimum verification level.

7.8 Enrollment

At the enrollment phase, the client registers biometrics that are compatible with the application built-in capabilities. For instance, if a mobile application is created for a specific integration that requires specific default biometrics, such integration could be hard-coded in the application.

Security Level		Requirement	Example
Highest	Genesis 4	Physical verification of a document proving identity along with the subject	Driver's license or passport verification
	Genesis 3	Verification of ownership of 2 or more digital assets	Both, Email + SMS verification
	Genesis 2	Verification of ownership of 1 digital asset that requires credentials	Email verification
Lowest	Genesis 1	Verification of ownership of 1 asset	SMS verification

(New)

Figure 12—Security levels of Genesis example

The BOPS platform addresses the speed of the biometric authentication transaction and solves the problem of a virtualized threat on a client device. Such a threat assumes that an intruder decompiles software on a copied virtual image of, for instance, a mobile device; uses this source code to stop authentication calls; and attempts to get control of the server that authenticates and grants permissions.

7.9 Biometric matching with visual cryptography

To mitigate these risks, BOPS encrypts the IBV without an encryption key, then stores one half of the IBV on the client device and the other half on the server. The IBV flow is visualized in Figure 13. The biometric matching occurs on the server. In this way, a stolen device cannot bypass authentication, hence a compromised device or server renders no useful information to an attacker.

The following assumptions are essential for establishing a processing agreement to biometrics authentication in BOPS:

- The biometric vector is split between the client and server.
- The approach to authentication does not depend on the source of the biometric.