Tutorial zur Verschlüsselung in Datenverteilersystemen

Jonathan Haas, Roland Schmitz

27. Juli 2016

Kappich Systemberatung

Inhaltsverzeichnis

1 Einleitung

In Version 3.9.0 der Kernsoftware wurde ein verbessertes Authentifizierungsprotokoll und eine Verschlüsselung der Verbindung zwischen Applikation und Datenverteiler sowie zwischen zwei Datenverteilern implementiert.

Wie das bisherige Protokoll, welches auf einem HMAC-Verfahren[1] basierte, bietet auch das neue Verfahren weiterhin folgende Sicherheitsmerkmale:

- Das Passwort wird nicht im Klartext übertragen.
- Schutz vor Replay-Angriffen[2]: Jemand, der den Datenverkehr bei der Authentifizierung mithört, gewinnt dadurch keine Informationen, die ihm später selbst die Authentifizierung ermöglichen.

Die Vorteile des neuen Verfahrens sind:

- Passwörter müssen nicht mehr im Klartext in der Konfiguration und in den Authentifizierungsdateien (passwd) abgespeichert werden.
- Durch die Verschlüsselung kann verhindert werden, dass ein Angreifer den Datenverkehr abhört oder unbemerkt manipuliert.
- Die benutzerverwaltung.xml kann gegen den Diebstahl der Passwörter abgesichert werden. In dieser Datei werden nur noch Überprüfungscodes gespeichert, mit denen ein Passwort auf Korrektheit geprüft werden kann, aber es ist nicht möglich, sich mit den Überprüfungscodes zu authentifizieren.
- Sobald ein Login erfolgreich durchgeführt wurde, kann eine Applikation oder ein gekoppelter Datenverteiler sich sicher sein, dass der Kommunikationspartner das korrekte Passwort (oder zumindest den richtigen Überprüfungscode) kennt. Es ist daher nicht mehr ohne weiteres möglich, einen Anwender zu einem gefälschten Datenverteilersystem umzuleiten, da sich beide Kommunikationspartner gegenseitig authentifizieren.
- Bisher authentifizierte sich die Konfiguration beim Datenverteiler, mit dem neuen Verfahren authentifiziert sich der Datenverteiler jetzt zusätzlich bei der Konfiguration. Dadurch kann die Gefahr verringert werden, dass sich die Konfiguration zu einem falschen Datenverteiler verbindet.

[1] https://de.wikipedia.org/wiki/Keyed-Hash_Message_Authentication_Code
[2] https://de.wikipedia.org/wiki/Replay-Angriff

- Perfect Forward Secrecy: Eine verschlüsselte Datenübertragung kann selbst dann nicht entschlüsselt werden, wenn das Passwort nachträglich einem Angreifer bekannt wird.

1.1 Grundlagen der Authentifizierung

Die Authentifizierung bei einem Datenverteiler kann wie bisher über die Angabe von Benutzername und Passwort erfolgen, zusätzlich können für den automatischen Start von Applikationen Login-Token eingesetzt werden, die die Anmeldung ohne Klartextpasswort erlauben. Zur Authentifizierung wird hierbei das Verfahren SRP (Secure Remote Password)[3] in Version 6[4] eingesetzt. Es handelt sich hierbei um ein Protokoll zur Passwort-basierten Authentifizierung, bei dem der Server (also Datenverteiler oder Konfiguration) das Klartextpasswort nie mitgeteilt bekommt, er aber dennoch über kryptographische Funktionen überprüfen kann, ob ein Client dieses Passwort weiß oder nicht. Details zum Ablauf der Authentifizierung finden sich auf der Website von NimbusSRP[5].

Zum Setzen eines verschlüsselten Passworts berechnet der Client aus dem Klartextpasswort und einem Zufallstext über ein kryptographisches Hashverfahren einen Login-Token. Aus diesem Login-Token lässt sich wiederum mit einem mathematischen Verfahren (welches auf diskreten Logarithmen basiert) ein Überprüfungscode berechnen, mit dem man überprüfen kann ob der Login-Token korrekt ist, aber aus dem dieser nicht wiederherstellbar ist. Der Überprüfungscode wird in der Konfiguration zusammen mit dem Zufallstext gespeichert.

Beim Login erzeugen Client und Server über ein mathematisches Schlüsselaustauschverfahren einen geheimen, zufälligen Sitzungsschlüssel, der bei Client und Server nur dann identisch ist, wenn der Überprüfungscode zum Login-Token passt. Der Client sendet aber weder Passwort noch Login-Token an den Datenverteiler oder die Konfiguration. Ist der Sitzungsschlüssel identisch, wird er zum Aufbau einer Verschlüsselung verwendet.

Wesentlich ist: Aus dem Passwort (und dem Zufallstext) ist ein Login-Token ableitbar, und aus dem Login-Token ein Überprüfungscode, aber aus dem Überprüfungscode ist kein Login-Token ableitbar und aus dem Login-Token kein Passwort. Der Client benötigt zum Einloggen mindestens den Login-Token, der Datenverteiler bzw. die Konfiguration benötigt zum Überprüfen des Passworts nur den Überprüfungscode.

Daraus folgt, dass in der lokalen Authentifizierungsdatei (passwd) statt den Klartextpasswörtern nur noch der Login-Token gespeichert werden muss, und in der konfigurationsseitigen benutzerverwaltung.xml nur noch der Überprüfungscode. Aus Kompatibilitätsgründen können aber die Login-Token und Überprüfungscodes von der Software

[3] https://en.wikipedia.org/wiki/Secure_Remote_Password_protocol
[4] https://tools.ietf.org/html/rfc5054
[5] http://connect2id.com/products/nimbus-srp/usage

automatisch berechnet werden, sofern die dafür notwendigen Grundlagen (wie das Klartextpasswort) gespeichert sind, daher kann das neue Authentifizierungsverfahren übergangsweise (bis ein neues Passwort vergeben wird) auch noch mit Klartextpasswörtern benutzt werden.

Tabelle 1.1: Unterstützte Authentifizierungsdaten in den Dateien

	Authentifizierungsdatei (passwd)	benutzerverwaltung.xml
Klartextpasswort	Ja, aus Kompatibilitätsgründen	Ja, aus Kompatibilitätsgründen
Login-Token	**Ja**	Nein
Überprüfungscode	Nein	**Ja**

1.2 Grundlagen der Verschlüsselung

Bei dem SRP-Schlüsselaustausch wird ein geheimer Schlüssel gebildet, mit dem die direkte Verbindung zwischen zwei Kommunikationspartnern verschlüsselt wird (Verfahren AES-GCM). Dadurch wird die Übertragung gegen das Mithören und gegen Manipulation abgesichert. Dies passiert automatisch und auch wenn noch Klartextpasswörter verwendet werden. Eine spezielle Konfiguration ist nicht erforderlich.

Daten die von einer Applikation gesendet werden, werden also in verschlüsselter Form an den Datenverteiler übertragen, dort entschlüsselt und für jeden auf diese Daten angemeldeten Kommunikationspartner erneut verschlüsselt, jedenfalls sofern dieser auch das neue Verfahren nutzt. Wenn eine Ende-zu-Ende Verschlüsselung benötigt wird, muss das über eine zusätzliche Verschlüsselung der Datentelegramme innerhalb der beiden beteiligten Applikation erfolgen. Beispielsweise wird die Übertragung von Benutzerverwaltungsanfragen zwischen Applikation und Konfiguration noch einmal zusätzlich Ende-zu-Ende verschlüsselt, da sonst die Gefahr besteht, dass eine (erfolgreich authentifizierte) Applikation die übertragenen Daten (wie Passwörter etc.) abhört.

Die Verschlüsselung der Verbindung ist abschaltbar, falls das aus Performancegründen notwendig sein sollte. Die meisten aktuellen Prozessoren besitzen aber spezielle Befehle für AES[6], sodass der Zusatzaufwand für die Verschlüsselung nicht ins Gewicht fallen sollte.

1.3 Glossar

Authentifizierungsdatei: Datei, die eine Applikation verwendet um darin Passwörter oder Login-Token für den automatischen Login zu speichern. Wird nicht benötigt, wenn ein interaktiver Login verwendet wird. Der Datenverteiler verwendet

[6] https://de.wikipedia.org/wiki/AES_(Befehlssatzerweiterung)

ebenfalls eine Authentifizierungsdatei um die Zugangsdaten zur Anmeldung bei der Konfiguration und bei anderen Datenverteilern zu speichern. Die Datei hat üblicherweise den Namen "passwd".

benutzerverwaltung.xml: Datei, in der die Konfiguration die Benutzerpasswörter bzw. Überprüfungscodes speichert und damit feststellen kann, ob sich ein Benutzer als Applikation oder gekoppelter Datenverteiler anmelden darf oder nicht.

Client: Die Applikation oder der Datenverteiler, der die Verbindung aufbaut. Wenn sich eine Applikation zum Datenverteiler verbindet, ist immer die Applikation der Client. Bei der Kopplung von 2 Datenverteilern wird konfiguriert, welcher der beiden Datenverteiler die Verbindung aufbauen soll, dieser ist dann der Client. Bei der SRP-Authentifizierung authentifiziert sich zuerst der Client beim Server. Bei der Verbindung von 2 Datenverteilern erfolgt danach eine zweite Authentifizierung in Gegenrichtung.

Interaktiver Login: Login, bei der ein Benutzername und Passwort manuell vom Anwender eingetippt wird. Sicherer als der automatische Login mit Authentifizierungsdateien, da nirgendwo ein Passwort oder Login-Token gespeichert werden muss.

Kryptographische Parameter: Die an einem Überprüfungscode gespeicherten kryptographischen Parameter, die die Sicherheit beeinflussen können (und bei zu hohen Werten auch die Authentifizierung und die Datenübertragung verlangsamen können). Standardmäßig werden die folgenden Parameter verwendet:

`t:96 L:128 H:SHA-256 KDF:PBKDF2WithHmacSHA256 c:20000 dkLen:256 n:1024 sLen:16`

Erklärung zu den einzelnen Werten:

t: Anzahl Bits zur Sicherstellung der Nachrichtenintegrität mit AES-GCM. Jedes verschlüsselte Telegramm wird mit einer Prüfsumme dieser Länge versehen. Empfohlener Minimum-Wert: 96

L: Anzahl Bits der Schlüssellänge der AES-Verschlüsselung. Mögliche Werte: 128, 192, 256. (Größere Werte als 128 benötigen bei der Oracle JVM möglicherweise die Unlimited Strength Java Cryptography Extension Policy Files[7]).

H: Auswahl der Hashfunktion in SRP unter anderem zur Berechnung von Sitzungsschlüsseln, beispielsweise "SHA–256".

KDF: Auswahl der Schlüsselableitungsfunktion zur Berechnung von Login-Token, beispielsweise "PBKDF2WithHmacSHA256".

c: Anzahl der Iterationen der Schlüsselableitungsfunktion zur Berechnung vom Login-Token. Empfohlener Minimum-Wert: 10000

[7] http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html

dkLen: Anzahl Bits für das Resultat der Schlüsselableitungsfunktion. Empfohlener Minimum-Wert: 128

n: Anzahl Bits der SRP-Primzahl N. Empfohlener Minimum-Wert: 1024

sLen: Anzahl Bytes für den SRP-Zufallstext (Salt). Empfohlener Minimum-Wert: 16

Die Parameter **t** und **L** können in der Benutzerverwaltung geändert werden, ohne dass der Überprüfungscode ungültig wird. Sie betreffen nur die Stärke der Verschlüsselung. Alle anderen Parameter können nur geändert werden, in dem ein neuer Überprüfungscode erzeugt wird, da sonst der bisherige Überprüfungscode nicht mehr zu den Parametern passt, was die Authentifizierung verhindert.

Login-Token: Binärer/hexadezimaler Code, der statt dem Passwort benutzt werden kann, um sich beim Datenverteiler zu authentifizieren. Dient dazu, um ihn in der Authentifizierungsdatei-Datei (passwd) zur Authentifizierung von automatisch gestarteten Applikationen zu verwenden ohne das Klartextpasswort speichern zu müssen. Beispiel-Zeile in der Datei:

```
Tester=SRP6~~~~ 46f981cc468a72d3726112e8f7b33
```

passwd: Der übliche Name der Authentifizierungsdatei, die die Klartextpasswörter oder Login-Token zum automatischen Anmelden von Applikationen enthält.

Server: Der Datenverteiler beim Entgegennehmen einer ankommenden Verbindung (von einer Applikation oder einem anderen Datenverteiler). Das Gegenstück zum Client.

SRP: Secure Remote Password, ein modernes Protokoll zur Authentifizierung mit Passwörtern. Wird in Version 6 eingesetzt und ist in RFC5054[8] beschrieben.

Überprüfungscode: Binärer/hexadezimaler Code, mit dem mit dem SRP-Verfahren geprüft werden kann, ob ein Benutzer das korrekte Passwort (bzw. den korrekten Login-Token) kennt. Wird in der benutzerverwaltung.xml statt dem Passwort eingetragen. Beispiel:

```
passwort="SRP6~~~~ v:51a927779f2e818dccc8a25fbc5d96227cf81b1df9dba6b
s:13a3cd77910c85cdd95dc13aa4d6d2b9 t:96 L:128 H:SHA-256
KDF:PBKDF2WithHmacSHA256 c:20000 dkLen:256 n:256 sLen:16"
```

Der hinter v: stehende Text ist hierbei der eigentliche Überprüfungscode, der hinter s stehende Text ist der zugehörige Zufallstext (Salt) und die restlichen Werte sind zusätzliche kryptographische Parameter.

[8] https://tools.ietf.org/html/rfc5054

1.4 Inhalt und Struktur der Dateien

1.4.1 Authentifizierungsdatei (passwd)

Die Authentifizierungsdatei wird von Client zum automatischen Login verwendet und kann bei Verwendung der neuen SRP-Authentifizierung entweder Klartextpasswörter oder Login-Token enthalten. SRP-Login-Token werden dabei durch den Präfix "SRP6∼∼∼∼" gekennzeichnet. Zur Unterscheidung dürfen Klartextpasswörter daher nicht 4 Tilden gefolgt von einem Leerzeichen enthalten.

Die Authentifizierungsdatei ist eine normale Textdatei. Eine Zeile besteht aus dem Benutzernamen, einem Gleichzeichen oder Doppelpunkt und dann dem Klartextpasswort oder Login-Token. Beispiel:

```
Konfiguration=SRP6~~~~ 7c36080fd21f2cfb4476897130885
Admin=SRP6~~~~ 46f981cc468a72d3726112e8f7b33
Tester=geheim
```

Der Datenverteiler verwendet ebenfalls eine Authentifizierungsdatei um die Passwörter für die Authentifizierung bei der lokalen Konfiguration sowie bei anderen Datenverteilern zu speichern. Es in der Authentifizierungsdatei vom Datenverteiler mit der neuen Version möglich, je Kommunikationspartner ein eigenes Passwort bzw. Login-Token anzugeben. Hierbei wird die Pid des Datenverteilers mit einem @-Zeichen an den Benutzernamen angehängt. Alternativ kann die Pid des Konfigurationsverantwortlichen angegeben werden, um das Passwort zur Authentifizierung bei der Konfiguration festzulegen. Beispiel:

```
Tester=SRP6~~~~ 46f981cc468a72d3726112e8f7b33
Tester@dav.TestUZ=KlartextPasswort
Tester@dav.VRZ3=SRP6~~~~ 3ef530a0f5a669fd2c652339485bf
Tester@kv.test=SRP6~~~~ 7c36080fd21f2cfb4476897130885
```

Ist für einen Kommunikationspartner kein spezifischer Eintrag vorhanden, wird das Standard-Passwort (ohne @-Suffix) verwendet (sofern vorhanden). Einfache Applikationen können die @-Syntax nicht verwenden.

1.4.2 benutzerverwaltung.xml

Der Aufbau der benutzerverwaltung.xml hat sich mit dem Update nur insofern geändert, als dass zusätzlich zu Klartextpasswörtern nun auch Überprüfungscodes im "passwort"-Attribut zugelassen sind. Eine Datei kann also beispielsweise so aussehen:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE benutzerkonten
PUBLIC "-//K2S//DTD Authentifizierung//DE" "authentication.dtd">
<benutzerkonten>
    <benutzeridentifikation
            admin="ja"
            name="Tester"
            passwort="SRP6~~~~ v:e5a5456965da2b48a852264f2aa… sLen:16">
        <autorisierungspasswort
                gueltig="ja"
                passwort="SRP6~~~~ v:922247a1a6f35a6f65befed… sLen:16"
                passwortindex="0"/>
    </benutzeridentifikation>
    <benutzeridentifikation
            admin="nein"
            name="Parametrierung"
            passwort="Klartextpasswort"/>
    <benutzeridentifikation
            admin="ja"
            name="Datenverteiler"
            passwort="SRP6~~~~ v:87348b3bf3d4bea44fe5f965ad4… sLen:16"/>
</benutzerkonten>
```

(Die Überprüfungscodes wurden zur besseren Darstellung gekürzt)

2 Migrationsschritte

2.1 Schritt 1: Benutzerobjekt für Datenverteiler anlegen

Mit der neuen Authentifizierung wird ein Benutzer benötigt, mit dem der Datenverteiler sich bei der Konfiguration authentifizieren kann. In vielen Fällen existiert dieser vermutlich bereits, falls nicht sollte er (um unnötige Neustarts zu vermeiden) bereits vor dem Softwareupdate angelegt werden. Wichtig ist:

- Der Benutzer des Datenverteiler wird mit `-benutzer=...` beim Start angegeben
- Der Benutzer ist als Systemobjekt in der Konfiguration vorhanden (`typ.benutzer`)
- Der Benutzer ist in der benutzerverwaltung.xml vorhanden
- Der Benutzer besitzt Admin-Rechte in der benutzerverwaltung.xml

Ein solcher Benutzer kann mit der Rahmenwerk-Benutzerverwaltung oder mit dem Migrationstool angelegt werden (eventuelle Warnungen, dass die Verschlüsselung nicht funktioniert, können dann ignoriert werden).

2.2 Schritt 2: Update der Kernsoftware

Voraussetzung für die Verwendung der Verschlüsselung ist das Update der Kernsoftware auf die Version 3.9.0 oder neuer. Sobald Datenverteiler, Konfiguration und Applikation die neue Kernsoftware-Version verwenden, wird die neue Authentifizierung und Verschlüsselung automatisch verwendet. Die Applikation signalisiert die verschlüsselte Verbindung durch eine Debug-Meldung ähnlich der folgenden:

```
INFO  : de.bsvrz.dav.daf.communication.protocol.ClientHighLevelCommunication
Verschlüsselte Verbindung aufgebaut mit: AES_128/GCM/NoPadding
```

Sind noch nicht alle Softwareeinheiten aktualisiert, wird weiterhin die alte Authentifizierung und keine Verschlüsselung verwendet. Falls bereits eine neue Applikation, aber noch ein alter Datenverteiler oder eine alte Konfiguration verwendet wird, wird eine entsprechende Warnung über die Debug-Funktionen erzeugt.

Das Rahmenwerk 2 verwendet die neue Kernsoftware automatisch, sobald die entsprechenden Plugins installiert sind. Hierfür sollte die Updatesite **Kernsoftware Gesamt V3.9.0** (oder ggf. neuer) von nerz-ev.de[1] heruntergeladen werden. Über den Rahmenwerk-Dialog *Neue Software installieren…* muss danach aus dieser Updatesite das Feature *Kernsoftware* installiert werden. Die erfolgreiche Installation der neuen Kernsoftware im Rahmenwerk kann geprüft werden, indem in den Debug-Ausgaben vom Rahmenwerk eine der folgenden Meldungen beim Verbindungsaufbau erscheint, je nachdem welche anderen SWE schon aktualisiert wurden:

- Verschlüsselte Verbindung aufgebaut mit: AES_128/GCM/NoPadding
- Der Datenverteiler unterstützt keine Verschlüsselung und sollte aktualisiert werden
- Verschlüsselte Anmeldung per SRP fehlgeschlagen, da die Konfiguration keine Verschlüsselung unterstützt

Damit die Meldungen ausgegeben werden, ist es ggf. sinnvoll das Rahmenwerk aus einer Kommandozeile heraus auszuführen.

2.3 Schritt 3: Passwörter für Applikationen aktualisieren

Sobald erfolgreich eine verschlüsselte Verbindung mit der neuen Softwareversion aufgebaut wurde, sollten neue Passwörter vergeben werden. Mit der neuen Software gesetzte Passwörter werden aus Sicherheitsgründen nicht mehr im Klartext in der Konfiguration gespeichert, sondern nur noch als Überprüfungscode.

Hierzu kann entweder das Migrationswerkzeug verwendet werden, oder die üblichen Werkzeuge der Benutzerverwaltung (beispielsweise im Rahmenwerk). Es sollte sichergestellt sein, dass im Rahmenwerk die aktuellen Kernsoftware-Plugins in Mindestversion 3.9.0 installiert sind.

Das zukünftige (manuelle) Einfügen von Klartextpasswörtern in die benutzerverwaltung.xml sollte vermieden werden.

2.3.1 Passwörter mit dem Migrationswerkzeug aktualisieren

Das Migrationswerkzeug kann entweder Online über die Datenverteilerverbindung arbeiten, oder Offline mit der benutzerverwaltung.xml.

Zum Starten des Migrationswerkzeugs muss die Hauptklasse `de.bsvrz.dav.daf.userManagement.UserManagement` gestartet werden, hierfür kann beispielsweise das mitgelieferte Skript verwendet werden.

[1] http://nerz-ev.de/produkte/software-dokumente

Das Tool greift standardmäßig auf eine `passwd`-Datei im aktuellen Ordner zu, um auf Wunsch die Login-Token zu speichern. Existiert eine solche Datei nicht, wird sie bei Bedarf automatisch angelegt. Über den Aufrufparameter `-authentifizierung=` kann ein anderer Dateiname angegeben werden.

Es erscheint beim Start folgendes Auswahlmenü:

```
Navigation
1: Benutzerverwaltung über den Datenverteiler
2: Benutzerverwaltung über benutzerverwaltung.xml
3: Fortgeschrittene Werkzeuge
q: Beenden
Auswahl:
```

Durch Eintippen der entsprechenden Zahl und Bestätigung mit Enter kann eine der möglichen Aktionen ausgewählt werden. Mit Option 1 kann sich das Tool mit der Konfiguration über einen Datenverteiler verbinden, mit Option 2 kann der Pfad einer lokal gespeicherten benutzerverwaltung.xml angegeben werden.

Je nachdem welche Option man auswählt muss man sich ggf. beim Datenverteiler authentifizieren oder den Dateipfad angeben. Bei einigen Optionen wird der Standardwert in Klammern angegeben, diesen kann man einfach durch Return bestätigen ohne vorher etwas einzugeben.

Alternativ kann man das Werkzeug auch mit dem Aufrufparameter

```
-offline=pfad/zur/benutzerverwaltung.xml
```

direkt im Offline-Modus starten, oder mit beispielsweise

```
-online
-benutzer=Tester
-authentifizierung=interaktiv
-datenverteiler=localhost:8083
```

im Online-Modus.

nach der eventuellen Authentifizierung und Anmeldung sollte man das folgende Menü sehen:

```
Navigation
1: Migrations-Assistent
2: Benutzerliste
3: Benutzer erstellen
4: Fortgeschrittene Werkzeuge
q: Beenden
Auswahl:
```

Mit Option 1 kann der aktuelle Migrationsstatus abgefragt werden, es werden alle Benutzer ausgegeben, bei denen noch ein Klartextpasswort gespeichert ist, sowie ggf. andere Warnungen und Probleme. Für die meisten Probleme schlägt das Werkzeug nach Auswahl des entsprechenden Problems eine Lösung vor.

Mit Option 2 können allgemein vorhandene Benutzer bearbeitet werden und deren Passwörter (verschlüsselt) neu gesetzt werden, wenn die automatischen Lösungsstrategien nicht ausreichen.

Mit Option 3 kann ein neuer Benutzer erstellt werden. (Im Offline-Modus wird nur der Eintrag in der benutzerverwaltung.xml angelegt, das eigentliche Benutzerobjekt muss separat versorgt werden.)

Unter Option 4 befinden sich verschiedene Werkzeuge um Überprüfungscodes und Login-Token manuell zu erzeugen. Sie funktionieren auch ohne Datenverteileranbindung bzw. lokale benutzerverwaltung.xml und sind evtl. bei komplexen Situationen mit mehreren Datenverteilern hilfreich.

Um neue Passwörter zu setzen sollte zuerst der Migrationsassistent gestartet werden. Danach sollte in der nacheinander jeder Benutzer, bei dem noch Probleme erkannt worden sind, korrigiert werden, beispielsweise in dem ein neues Passwort vergeben wird. Die ideale Vorgehensweise unterscheidet sich je nach Art der Applikation:

```
Navigation
1: Neues Passwort setzen
2: Ab jetzt Klartextpasswörter setzen
3: Zufallspasswort für automatischen Login setzen
4: Überprüfungscode setzen
5: Login-Token für Authentifizierungsdatei erzeugen
6: Administrator-Rechte setzen
7: Passwort auf Korrektheit überprüfen
8: Authentifizierungstoken auf Korrektheit überprüfen
9: "Einmalpasswörter bearbeiten
10: Benutzer löschen
0: Zurück
q: Beenden
```

Die einzelnen Optionen bewirken folgende Aktionen: * Mit “Neues Passwort setzen” wird für einen vorhandenen Benutzer das Passwort geändert. * Über “Ab jetzt Klartextpasswörter setzen” kann im Ausnahmefall eingestellt werden, dass ab jetzt für die weiteren Aktionen wieder Klartextpasswörter an die Konfiguration übertragen werden. Dies ist nur für den Fall anzuwenden, wenn versehentlich für einen Benutzer zu früh auf Verschlüsselung umgestellt wurde. * Über “Ab jetzt verschlüsselte Passwörter setzen” kann wieder auf den Normalfall umgestellt werden. Die Option wird eingeblendet, wenn vorher auf Klartextpasswörter umgestellt wurde. * Über “Zufallspasswort für automatischen Login setzen” wird für Benutzer ein zufälliger Login-Token erzeugt. Ein interaktiver Login ist dann für den Benutzer nicht möglich. * Über “Überprüfungscode setzen” kann ein Überprüfungscode für einen Benutzer im laufenden Betrieb in der Konfiguration gespeichert werden (ohne das Klartextpasswort zu kennen). * Über “Login-Token für Authentifizierungsdatei erzeugen” wird ein Login-Token für die Authentifizierungsdatei erzeugt. * Über “Administrator-Rechte setzen” können die Administrator-Rechte gegeben bzw. wieder entzogen werden. * Über “Passwort/Authentifizierungstoken auf Korrektheit überprüfen” wird die Eingabe entsprechend geprüft. * Über “Einmalpasswörter bearbeiten” können für den Benutzer die Einmalpasswörter bearbeitet werden. * Über “Benutzer löschen” wird der entsprechende Benutzer gelöscht.

2.3.1.1 Bediener

Für Benutzer, die dazu gedacht sind, in Bedienungen, grafischen Programmen und ähnlichem zum Login verwendet zu werden, sollte mit Option 1 ein neues Passwort gesetzt werden. Falls dieser Benutzer bereits in der Authentifizierungsdatei eingetragen ist, bietet das Tool an, einen passenden Login-Token zu erzeugen und das bisherige Passwort in der Authentifizierungsdatei zu ersetzen. Wählt man hier Nein (n) enthält man die Option, das bisherige Passwort aus der Datei zu löschen. Das ist generell die empfohlene Vorgehensweise da so verhindert wird, dass jemand das Passwort oder den Login-Token aus der Datei stehlen kann. Die neue Kernsoftware unterstützt seit 3.9.0 bei jedem Programm, welches die Standardbibliothek verwendet, die interaktive Passworteingabe über den Aufrufparameter “-authentifizierung=interaktiv”.

2.3.1.2 Automatisch gestartete Programme

Für Benutzer, die für automatisch gestartete Programme (Archiv, Parametrierung, etc.) verwendet werden kann mit Option 2 ein zufälliger Login-Token erzeugt werden. Dieser wird automatisch in die passwd-Datei eingetragen. Ein (bekanntes) Klartextpasswort zu diesem Benutzer gibt es dann nicht.

Tabelle 2.1: Bisherige Orte für die Speicherung der Passwörter für spezielle Applikationen

	Authentifizierungsdatei (passwd)	benutzerverwaltung.xml
Datenverteiler	Erforderlich	Optional, in der Regel aber vorhanden
Konfiguration	Erforderlich	-
Parametrierung	Erforderlich	-

Tabelle 2.2: Neue Orte für die Speicherung der Passwörter für spezielle Applikationen

	Authentifizierungsdatei (passwd)	benutzerverwaltung.xml
Datenverteiler	Erforderlich	Erforderlich
Konfiguration	Erforderlich	-
Parametrierung	Nur wenn nicht in benutzerverwaltung.xml	Empfohlen

2.3.1.3 Parametrierung

Für die Parametrierung, die in früheren Kernsoftwareversionen eine Sonderstellung hatte, kann nun ebenfalls ein eigener Benutzer in der Konfiguration erstellt werden, wodurch das Passwort wie bei normalen Applikationen auch durch die Konfiguration geprüft wird. Falls die Parametrierung eine andere passwd-Datei benutzt als der Datenverteiler kann dann auch in der passwd-Datei vom Datenverteiler das Passwort für den Parametrierungsbenutzer entfernt werden.

2.3.1.4 Konfiguration und Datenverteiler

Die Authentifizierung zwischen Konfigurations und Datenverteiler läuft mit der neuen Software etwas anders ab, als bisher. Der Datenverteiler sollte in der Konfiguration einen eigenen Benutzer definiert bekommen. Dieser muss in der benutzerverwaltung.xml als Administrator markiert sein, damit der Datenverteiler berechtigt ist, Benutzer zu authentifizieren. Wie bei den anderen Applikationen auch sollte in der Konfiguration dann ein Überprüfungscode gespeichert werden und in der Datenverteiler-passwd-Datei ein Login-Token. Das Vorgehen ist hierbei exakt wie bei den anderen automatisch gestarteten Programmen.

Zusätzlich authentifiziert sich die Konfiguration wie bisher beim Datenverteiler. Dabei verwendet die Konfiguration einen Pseudo-Benutzer, den es nicht in der benutzerverwaltung.xml geben muss bzw. sollte. Wichtig ist hier nur, dass in der passwd-Datei von Konfiguration und Datenverteiler (falls nicht beide die gleiche Datei verwenden) das selbe Passwort steht. Hierbei kann das Passwort durch ein Login-Token ersetzt werden. Hierzu kann man beispielsweise im Migrationstool “Fortgeschrittene Werkzeuge”→“Zufälligen Login-Token erzeugen” aufrufen.

2.3.2 Passwörter mit den Datenverteilerfunktionen aktualisieren

Anstelle des Migrationswerkzeuges können die Passwörter auch über die Schnittstelle `DataModel.getUserAdministration()` aktualisiert werden, bzw. über GUI-Werkzeuge, die diese Schnittstelle verwenden. Sofern alle beteiligten Programme die Verschlüsselung unterstützen, wird dann automatisch nur noch der Überprüfungscode übertragen und in der Konfiguration gespeichert.

Wichtig: egal, wie das Passwort aktualisiert wurde, sobald in der benutzerverwaltung.xml ein verschlüsseltes Passwort (SRP-Überprüfungscode) gespeichert wurde wird bei Verwendung mit dem alten Verfahren bzw. der alten Software der Login verweigert (Meldung: "Die Authentifikationsdaten sind fehlerhaft."). Die Kernsoftware sollte also vor einer Passwortänderung auf allen Systemen aktualisiert werden, von denen aus sich der Benutzer einloggen können muss. Wenn noch nicht alle Systeme umgestellt sind, kann es ggf. sinnvoll sein eine Passwortänderung entweder zu verschieben bis alle Systeme aktualisiert sind, die Änderung im Klartext manuell in der benutzerverwaltung.xml durchzuführen oder vor der Passwortänderung die Umgebungsvariable (Unterabschnitt 5.1.1) `srp6.disable.verifier` zu setzen, was dazu führt, dass das Passwort im Klartext übertragen und gesetzt wird.

2.4 Schritt 4: Passwörter für gekoppelte Datenverteiler aktualisieren

Auch die Passwörter für die Datenverteilerkopplung sollten auf das neue Verfahren umgestellt werden, sobald beide gekoppelten Systeme aktualisiert wurden.

Dieser Schritt kann prinzipiell auch gleichzeitig mit Schritt 2 oder davor gemacht werden und ist vom Verfahren her eigentlich identisch, jedoch ergeben sich zusätzliche Komplikationen dadurch, dass der Überprüfungscode für das eigene Passwort auf einem fremden System gespeichert werden muss. Grundsätzlich gibt es mehrere Möglichkeiten, den neuen Überprüfungscode auf dem fremden System zu speichern:

- Wenn die fremde Konfiguration über den Datenverteiler normal erreichbar ist und die nötigen Rechte vorhanden sind kann man über das Migrationswerkzeug online (bei Eingabe des fremden Konfigurationsverantwortlichen) oder mit den normalen Benutzerverwaltungsfunktionen im Datenverteiler sein Passwort in der fremden Konfiguration normal ändern.
- Mit dem Migrationswerkzeug manuell durch die Funktion "Fortgeschrittene Werkzeuge"→"Login-Token und Überprüfungscode berechnen" einen Login-Token und Überprüfungscode aus einem eingegebenen Passwort berechnen. Der Überprüfungscode kann per E-Mail oder ähnlich an den Administrator des fremden Systems gesendet werden, der ihn in der benutzerverwaltung.xml einträgt und der

Login-Token wird lokal in der Datenverteiler-Authentifizierungsdatei (passwd) gespeichert (idealerweise mit der Datenverteiler-Pid des fremden Systems als Suffix), beispielsweise:

```
Tester@dav.VRZ3=SRP6~~~~ 3ef530a0f5a669fd2c652339485bf
```

- Das Passwort und den Überprüfungscode auf dem Zielsystem erzeugen und in die Konfiguration schreiben (analog zu normalen Applikationen) und das Klartextpasswort und/oder den Login-Token über einen sicheren Weg an denjenigen schicken, der sich authentifizieren soll.

3 Sonstige Hinweise

3.1 Sicherheit

Um die Sicherheit des Datenverteilersystems zu gewährleisten müssen folgende Richtlinien eingehalten werden:

- Alle Passwörter, die im Klartext in Dateien standen, sollten als unsicher betrachtet werden und durch neue Überprüfungscodes oder Login-Token ersetzt werden, denen ein anderes, sicheres Klartextpasswort zugrunde liegt.
- Es sollten keine Klartextpasswörter mehr manuell in die benutzerverwaltung.xml oder die Authentifizierungsdateien (passwd) geschrieben werden.
- Die Software selbst legt keine speziellen Passwortrichtlinien fest, außer dass Passwörter nicht leer sein dürfen. Obwohl das verwendete Verfahren auch eine relativ hohe Sicherheit bei schwächeren Passwörtern bietet, sollten bei der Vergabe von Passwörtern sinnvolle Richtlinien eingehalten werden. Die Verwendung von zxcvbn4j[1] oder gleichwertiger Software zur Bewertung von Passwortsicherheit innerhalb von GUI-Applikationen, die eine Passwortänderung anbieten, wird empfohlen.
- Authentifizierungs-Dateien sollten nur noch da benutzt werden, wo es für den automatischen Start von Applikationen erforderlich ist. Applikationen, die automatisch gestartet werden, sollten keine unnötigen Rechte erhalten und in der Konfiguration nicht als Administrator markiert sein. Der in der Authentifizierungsdatei gespeicherte Login-Token ist nur wenig sicherer als das Klartextpasswort und muss so geheim gehalten werden, wie möglich.
- Die Passwörter für die speziellen Benutzer für Konfiguration und Parametrierung dürfen nicht mehr die Standard-Passwörter (“configuration”, “parameter”) verwenden, das Passwort für die Konfiguration kann durch einen zufälligen Login-Token ersetzt werden (mit dem Migrationswerkzeug unter “Fortgeschrittene Werkzeuge” erstellbar)
  und das Passwort für die Parametrierung kann entweder auf die gleiche Weise ersetzt werden, oder besser (wie bei anderen automatisch gestarteten Applikationen auch) als Überprüfungscode in der Konfiguration gespeichert werden (ggf. vorher Benutzer für Parametrierung anlegen). Da für diese beiden Applikationen keine

[1] https://github.com/nulab/zxcvbn4j

Rechteprüfung durchgeführt wird, ist es besonders wichtig, dass hier sichere Passwörter verwendet werden und diese geheim gehalten werden.

- Die Rechte der benutzerverwaltung.xml und der Authentifizierungsdateien sollte so gesetzt werden, dass die Lese- und Schreibrechte soweit wie möglich eingeschränkt sind und keine unberechtigten Zugriffe möglich sind.
- Sobald alle Applikationen und Anwender umgestellt sind, sollte die alte Authentifizierung deaktiviert werden (Abschnitt 3.4).
- Falls die Überprüfungscodes in der benutzerverwaltung.xml unberechtigt ausgelesen wurden, besteht erst einmal kein Grund zur Panik, denn mit diesen kann man sich nicht authentifizieren. Es sind jedoch Brute-Force-Angriffe auf die Passwörter möglich, die schneller durchgeführt werden können, als wenn man sich für jeden Versuch beim Datenverteiler authentifizieren muss. Außerdem kann damit ein Angreifer möglicherweise vorgeben, der echte Datenverteiler bzw. die echte Konfiguration zu sein und damit beispielsweise übertragene Daten abfangen. Daher sollten so schnell wie möglich neue Passwörter vergeben werden.
- Falls Klartextpasswörter oder Login-Token verloren gehen, sind sofort die Passwörter und Login-Token zu ändern.

3.2 Einmalpasswörter

Mit der neuen Kernsoftwareversion funktionieren Einmalpasswörter anders als bisher. Während in frühen Kernsoftware-Versionen immer nur das nächste Einmalpasswort verwenden konnte und mit neueren Kernsoftware-Versionen ein beliebiges gültiges Einmalpasswort benutzt werden konnte, werden bei Verwendung der neuen Authentifizierung nun die Einmalpasswörter durchnummeriert.

Der Index des Einmalpassworts, der bisher keine Bedeutung hatte, wird nun benötigt, um sich einzuloggen.

3.2.1 Anlegen von Einmalpasswörtern

Mit der Umstellung auf SRP sollten die alten Einmalpasswörter gelöscht werden und neue vergeben werden. Hierfür kann das Migrationswerkzeug benutzt werden. “Benutzerliste”→*Benutzername*→“Einmalpasswörter bearbeiten”:

```
Einmalpasswörter:
Anzahl gültige Einmalpasswörter: 10

Navigation
```

```
1: Alle Einmalpasswörter löschen
2: Unverschlüsselte Einmalpasswörter löschen
3: Einmalpasswörter hinzufügen
4: Zufällige Einmalpasswörter hinzufügen
5: Gültige Einmalpasswörter auflisten
0: Zurück
q: Beenden
```

Über die Option "Gültige Einmalpasswörter auflisten" können die aktuell gültigen Passwort-Indizes aufgelistet werden, sowie der Status ob die Passwörter verschlüsselt gespeichert sind oder nicht.

Falls unverschlüsselte Passwörter vorhanden sind, können über die Option 1 oder 2 entweder alle Passwörter oder nur alle unverschlüsselten gelöscht werden.

Über die Option 3 können dann neue Passwörter manuell hinzugefügt werden, über die Option 4 können automatisch zufällige Einmalpasswörter generiert werden. Hierbei wird dann eine Passwortliste ähnlich einer TAN-Liste beim Online-Banking ausgegeben:

```
Anzahl: (10)
Neue Einmalpasswörter:
Tester-0=eonv-z48y-96wn-34ka
Tester-1=fggz-gb4d-o7bc-j9w8
Tester-2=rehr-h6ys-xzcr-dfog
Tester-3=fmh4-pdhg-whid-s2yy
Tester-4=n6z3-pt4z-p92e-3ydf
Tester-5=2x43-49jd-8fb6-enge
Tester-6=6km4-ucys-48qz-a9b2
Tester-7=v6ck-8gdc-o5ya-nrgi
Tester-8=hbdv-ibfn-y2t9-8tui
Tester-9=qrds-es5i-osp3-2rcu
```

Der Index des Einmalpassworts befindet sich hier zwischen Benutzernamen und dem Gleichzeichen(=), das Passwort befindet sich nach dem Gleichzeichen. Die Daten könnten theoretisch wie sie sind in eine Authentifizierungsdatei eingefügt werden, auch wenn diese eher für automatische Logins gedacht sind. Besser ist es vermutlich, die Passwortliste auszudrucken und sicher aufzubewahren.

Einmalpasswörter lassen sich nur für die Anmeldung von Applikationen nutzen und nicht für die Benutzerverwaltungs-Schnittstelle oder für die Kopplung von Datenverteilern.

3.2.2 Verwendung von Einmalpasswörtern

Die Einmalpasswörter können in einer Liste ähnlich wie eine TAN-Liste beim Online-Banking verwaltet werden. Jeder Index kann nur einmal verwendet werden. Beim Login muss der Index des Einmalpassworts mit einem Minuszeichen an den Benutzernamen angehängt werden. Will man sich beispielsweise mit der oben genannten Liste einloggen, wählt man beispielsweise folgende Login-Daten:

Benutzername: Tester–0

Passwort: eonv-z48y–96wn–34ka

Es gibt hier keine Verpflichtung mit dem ersten Index (hier 0) zu beginnen. Nach dem erfolgreichen Login sollte die Zeile gelöscht oder durchgestrichen werden, da das Passwort nicht mehr nutzbar ist.

Es ist mit der neuen Kernsoftware mit dem neuen Authentifizierungsverfahren nicht mehr möglich, sich mit einem Einmalpasswort einzuloggen ohne den Passwortindex anzugeben.

3.3 Deaktivierung der Verschlüsselung

Datenverteiler und Applikationen besitzen einen neuen Startparameter, mit dem die Verschlüsselung auf Wunsch deaktiviert werden kann. Ein Grund hierfür können Performanceprobleme auf sehr alten oder stark belasteten Systemen sein, bei denen die CPU keine speziellen Befehle für die AES-Verschlüsselung besitzt:

```
-verschluesselung=[immer,automatisch,nein]
```

Für Applikationen ist der Standardwert hierbei “immer”, für Datenverteiler ist er “automatisch”.

immer bedeutet, dass die Verschlüsselung in jedem Fall bei der neuen Authentifizierung aktiviert wird und bleibt.

automatisch bedeutet, dass die Verschlüsselung dann deaktiviert wird, wenn beide Verbindungspartner sich auf dem selben Rechner befinden und der andere Verbindungspartner die Verschlüsselung auch deaktivieren will.

nein bedeutet, dass dieser Verbindungspartner die Verschlüsselung auf jeden Fall deaktivieren will, aber damit sie wirklch deaktiviert wird, muss auch der andere Verbindungspartner damit einverstanden sein (also ebenfalls “nein” oder bei lokaler Verbindung “automatisch” gesetzt haben).

Will man daher bei allen oder nur bestimmten lokalen Applikationen die Verschlüsselung ausschalten, muss man bei den lokalen Applikationen den Aufrufparameter

```
-verschluesselung=automatisch
```

setzen. Der Datenverteiler braucht dann nicht konfiguriert zu werden, da er bereits `automatisch` als Standard verwendet. Will man die Verschlüsselung komplett abschalten muss sowohl bei Applikationen als auch beim Datenverteiler

```
-verschluesselung=nein
```

gesetzt werden.

3.4 Erzwingen der neuen Authentifizierung

Sobald alle beteiligten System auf die neue Software umgestellt sind, sollte die neue Authentifizierung erzwungen werden, was die Sicherheit erhöht, da damit verhindert werden kann, dass ein Angreifer die sichere SRP-Authentifizierung durch eine unsichere Verschlüsselung ersetzt. Hierfür sollte bei **allen Applikationen und beim Datenverteiler** der Startparameter

```
-erlaubeHmacAuthentifizierung=nein
```

gesetzt werden. Danach kann die alte unverschlüsselte Authentifizierung nicht mehr verwendet werden. In einem späteren Release der Kernsoftware wird das Default-Verhalten so geändert werden, dass standardmäßig nur noch die sichere Authentifizierung möglich sein wird und man zur Verwendung der alten Authentifizierung explizit

```
-erlaubeHmacAuthentifizierung=ja
```

angeben muss.

4 Anwendungsbeispiel

Im folgenden Abschnitt wird die Migration auf verschlüsselte Passwörter an der folgenden Beispielumgebung durchgeführt:

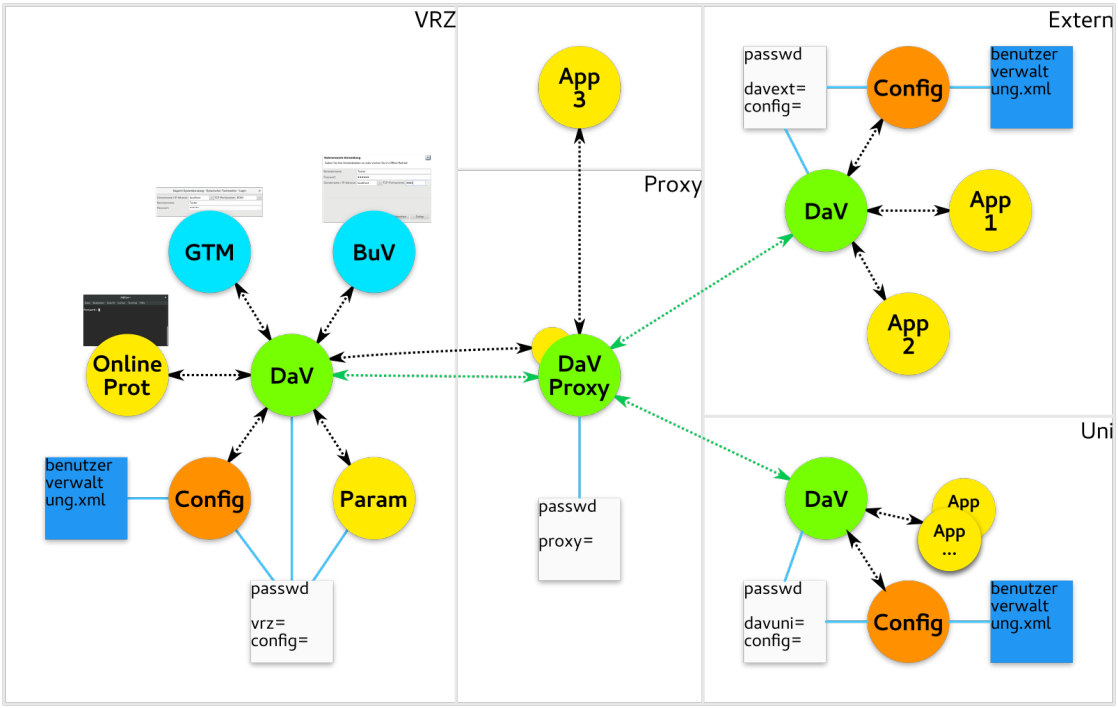

Abbildung 4.1: Übersicht Gesamtsystem

Bei dem Beispielsystem handelt es sich um eine VRZ, an die über einen Proxydatenverteiler verschiedene externe Systeme und weitere externe Applikationen angeschlossen sind. Die Umstellung erfolgt in mehreren Schritten, wobei in den ersten Schritten nur die VRZ und der Proxydatenverteiler umgestellt werden, ohne dass die externen Partner tätig werden müssen.

Dazu werden die folgenden Schritte durchgeführt:

1. Ggf. Anlegen von einem Benutzer für den Datenverteiler

- Um unnötige Neustarts zu vermeiden bietet es sich an, bereits jetzt einen eigenen Benutzer für den Datenverteiler anzulegen, sofern noch keiner existiert. Mit der neuen Authentifizierung benötigt der Datenverteiler einen Benutzer, unter dem er sich bei der Konfiguration authentifiziert. Hierfür wird der Benutzer verwendet, der beim Aufrufargument `-benutzer=` des Datenverteilers angegebenen ist. Fehlt dieses Aufrufargument, sollte es ergänzt werden.
- Der Benutzer des Datenverteilers muss in der Benutzerverwaltung als Administrator markiert sein.

2. Aktualisierung der Software und Neustart

   - Im Beispiel werden jetzt die Systeme VRZ und Proxy gleichzeitig beendet, die Kernsoftware auf 3.9.0 aktualisiert und wieder gestartet. Vorher bietet sich ein Backup an. Es wäre natürlich auch möglich, die Systeme einzeln zu aktualisieren. Beim Rahmenwerk müssen (spätestens vor dem Ändern der Passwörter) die neuen Kernsoftware-Plugins bzw. Features installiert werden.

3. Migrationswerkzeug starten

   - Sobald das System wieder läuft, wird das Migrationswerkzeug gestartet und online mit dem Datenverteilersystem verbunden. Über den Menü-Punkt *Migrations-Assistent* werden alle Benutzer aufgelistet, deren Passwort noch nicht umgestellt ist, sowie andere Probleme.

4. Änderung des Passworts für Konfiguration

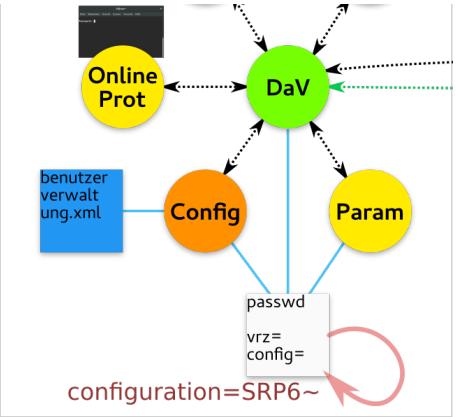

Abbildung 4.2: Passwort Konfiguration

- Die Konfiguration authentifiziert sich beim Datenverteiler über eine `passwd`-Datei, wobei der Datenverteiler dieselbe `passwd`-datei ausliest um das passwort zu überprüfen. Es reicht also, mit dem Migrationswerkzeug einen zufälligen Login-Token für die Konfiguration zu erzeugen und das Klartextpasswort in der passwd-Datei zu ersetzen.

Prüfung: In der `passwd`-Datei steht beim Benutzer *configuration* (bzw. dem von der Konfiguration verwendetem Benutzer) kein Klartextpasswort mehr, sondern der erzeugte Login-Token.

5. Änderung des Passworts für Datenverteiler

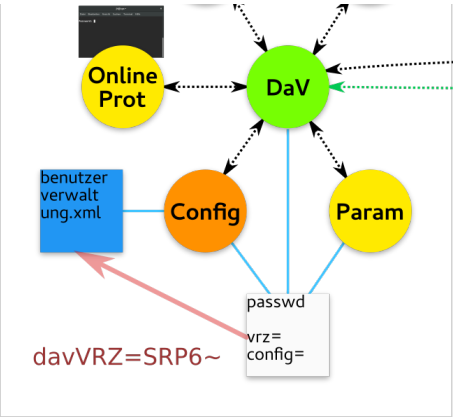

Abbildung 4.3: Passwort Datenverteiler

- Mit der neuen Authentifizierung authentifiziert sich neuerdings der Datenverteiler ebenfalls bei der Konfiguration. Hierzu sollte für den (ggf. im ersten Schritt erzeugten Datenverteilerbenutzer) ebenfalls ein zufälliger Login-Token gesetzt werden.

  Achtung: Wenn der Datenverteiler-Benutzer auch an anderen Stellen zur Authentifizierung verwendet wird, muss in den dortigen `passwd`-Dateien ebenfalls der neue Login-Token eingefügt werden. Alle diese Systeme müssen dann die neue Software verwenden.

  Prüfung: In der `passwd`-Datei steht beim Datenverteiler-Benutzer kein Klartextpasswort, sondern der erzeugte Login-Token. In der benutzerverwaltung.xml der *VRZ* steht ein (passender) Überprüfungscode.

6. Änderung des Passworts für Parametrierung und andere automatisch gestartete Applikationen

   - Für automatisch gestartete Applikationen wird genauso ein neues Passwort bzw. ein neuer Login-Token gesetzt, wobei der Überprüfungscode des verwendeten Benutzers in der benutzerverwaltung.xml eingefügt wird und der Login-Token in die `passwd`-Datei. Das Migrationstool erledigt das auf Wunsch automatisch.

     Der Benutzer für die Parametrierung benötigt theoretisch keinen Eintrag in der benutzerverwaltung.xml, da der Datenverteiler hier als Spezialfall (wie bei der Konfiguration) den Login mit der lokalen `passwd` überprüfen kann.

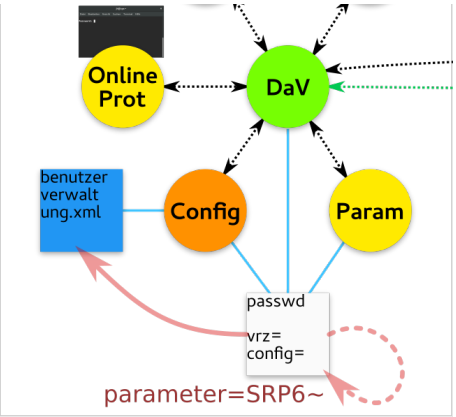

Abbildung 4.4: Passwort Parametrierung

Es ist aber sinnvoll, den Benutzer für die Parametrierung anzulegen und in der benutzerverwaltung.xml einzutragen.

Achtung: Wenn die Benutzer auch an anderen Stellen zur Authentifizierung verwendet wird, muss in den dortigen `passwd`-Dateien ebenfalls der neue Login-Token eingefügt werden. Alle diese Systeme müssen dann die neue Software verwenden.

Prüfung: In der `passwd`-Datei steht bei den betreffenden Benutzern kein Klartextpasswort, sondern der erzeugte Login-Token. In der benutzerverwaltung.xml der *VRZ* stehen (passende) Überprüfungscodes.

7. Einrichtung für interaktive Applikationen

   - Für interaktiv gestartete Applikationen (im Beispiel Online-Protokollierer, GTM, Rahmenwerk) sollte ebenfalls für die entsprechenden Benutzer neue Passworte erzeugt werden, wodurch der Überprüfungscode in der benutzerverwaltung.xml gespeichert wird. Im Gegensatz zu automatisch gestarteten Applikationen wird für diese Benutzer in der `passwd`-Datei kein Login-Token ergänzt bzw. das vorhandene Passwort entfernt. Im Startskript des Online-Protokollierers und anderer manuell gestarteter Kommandozeilen-Applikationen wird statt den ggf. vorhandenen `-authentifizierung=passwd` jetzt `-authentifizierung=interaktiv` gesetzt, wodurch das Passwort interaktiv abgefragt wird.

     Prüfung: In der `passwd`-Datei sind keine Einträge zu den Benutzern vorhanden. In der benutzerverwaltung.xml der stehen (passende) Überprüfungscodes und keine Klartextpasswörter. Der Login funktioniert noch.

8. Aktualisierung der Passwörter von gekoppelten Datenverteilern an der gleichen Konfiguration

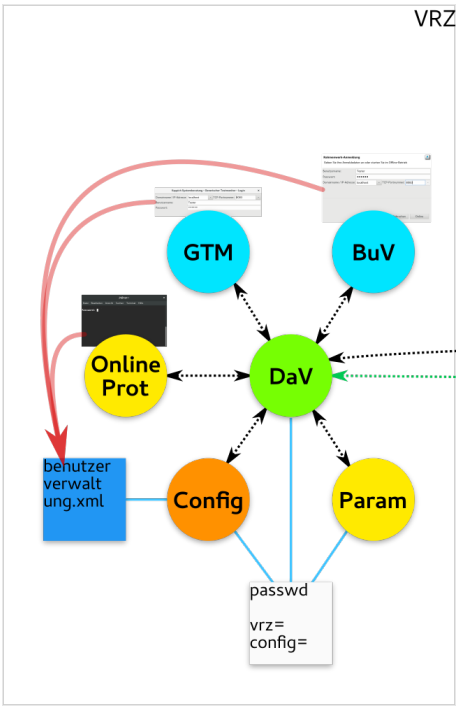

Abbildung 4.5: Interaktiver Login

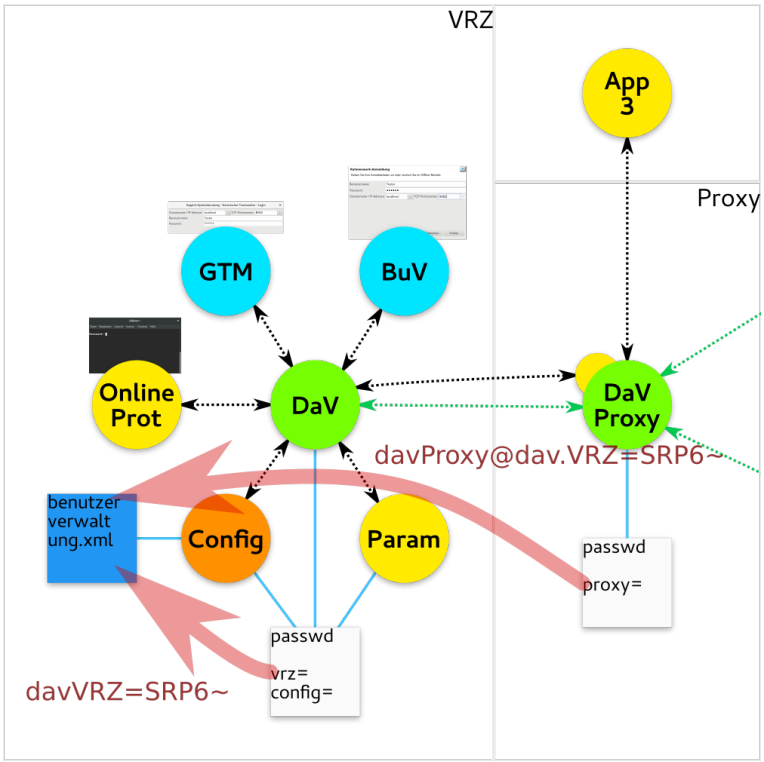

Abbildung 4.6: Passwort Proxy

- Der Proxy-Datenverteiler verwendet die gleiche Konfiguration und Benutzerverwaltung wie die VRZ auch. Beide Datenverteiler authentifizieren sich gegenseitig. In den `passwd`-Dateien sind die Login-Token zur authentifizierung beim jeweils anderne Datenverteiler zu setzen, die Konfiguration der VRZ speichert beide Überprüfungscodes. Das Passwort für den Benutzer `dav.VRZ` (unter diesem Benutzer authentifiziert sich der Datenverteiler der VRZ gegenüber dem Proxy-Datenverteiler) wurde hier schon in Schritt 5 verschlüsselt und braucht nicht noch einmal angepasst zu werden.

  Achtung: Wenn der Benutzer, mit dem der Proxy-Datenverteiler sich beim VRZ-Datenverteiler authentifiziert auch bei der Authentifizierung auf anderen (noch nicht umgestellten) Systemen benutzt wird, dann kann mit der `@`-Syntax wie in der Abbildung nur das Passwort zur Authentifizierung in Richtung *VRZ* geändert werden.

  Prüfung: In der `passwd`-Datei sind keine Einträge zu den Benutzern vorhanden. In der benutzerverwaltung.xml der stehen (passende) Überprüfungscodes und keine Klartextpasswörter. Der Login funktioniert noch.

9. Aktualisierung der Passwörter von externen gekoppelten Datenverteilern

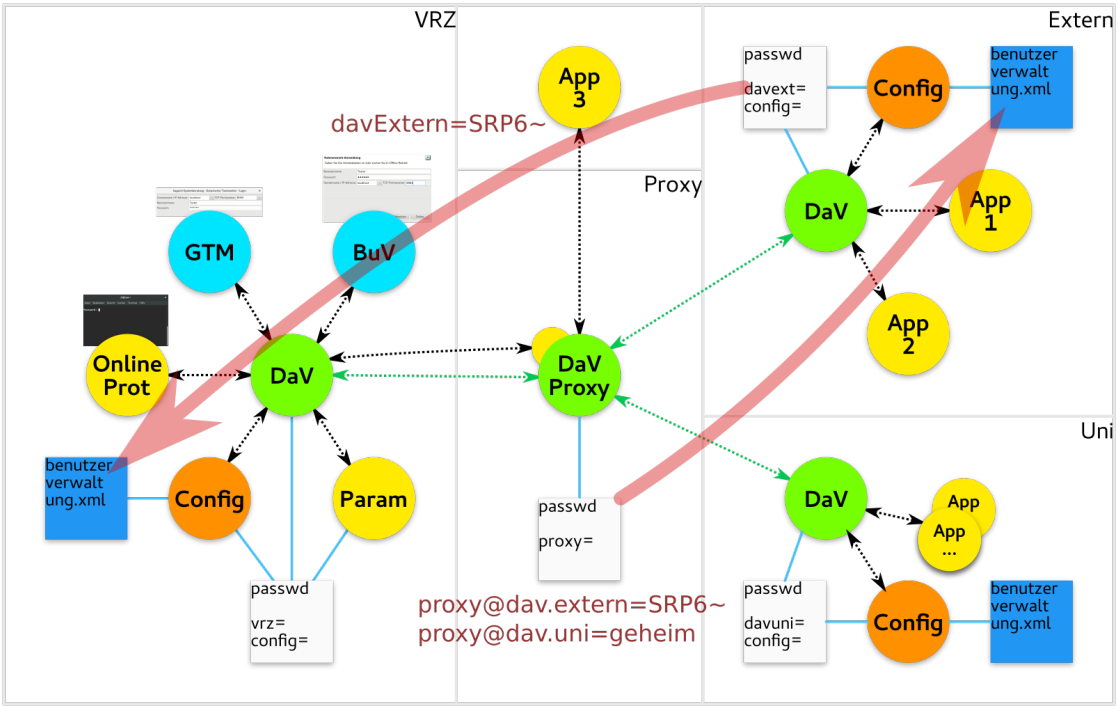

Abbildung 4.7: Passwort Extern

- Hier wurde nun auch die Software des *Extern*-Systems aktualisiert, wodurch auch die Passwörter für diese Verbindung verschlüsselt werden können. Das Vorgehen ist ähnlich zum vorherigen Abschnitt, allerdings werden hier unterschiedliche Konfigurationen verwendet. Der Überprüfungscode für den Benutzer *proxy* wird von dem VRZ-System in die externe Konfiguration übertragen. Gleichzeitig wird im externen System ein Paar aus Login-Token und Überprüfungscode erzeugt und der Überprüfungscode in der VRZ-Konfiguration gespeichert, sowie der Login-Token in der externen `passwd`-Datei.

  Achtung: Mit der @-Syntax in der `passwd` kann der Proxy unterschiedliche Passwörter je Verbindung definieren und so z.B. auch noch Klartextpasswörter speichern wenn noch nicht alle gekoppelten Systeme umgestellt sind.

  Prüfung: In der `passwd`-Datei sind keine Einträge zu den Benutzern vorhanden. In der benutzerverwaltung.xml der stehen (passende) Überprüfungscodes und keine Klartextpasswörter. Der Login funktioniert noch.

10. Überprüfung der Migration

    Am Schluss ist folgendes zu prüfen:

    - Das System ist funktionsfähig
    - Der Migrationsassistent zeigt keine Warnungen mehr an
    - Keine Klartextpasswörter und unnötigen Login-Token in `passwd`-Dateien vorhanden
    - Keine mit der Verschlüsselung oder Authentifizierung zusammenhängenden Warnungen in den Debug-Ausgaben von Datenverteiler und Konfiguration

5 Anhang

5.1 Konfiguration

5.1.1 Umgebungsvariablen

`srp6.params`: Über den JVM-Parameter `-Dsrp6.params="…"` bzw. eine entsprechend gesetzte Umgebungsvariable können die Details der Authentifizierung und Verschlüsselung konfiguriert werden. Das Format der Variablen entspricht folgendem Beispiel:

```
t:96 L:128 H:SHA-256 KDF:PBKDF2WithHmacSHA256 c:20000 dkLen:256 n:1024 sLen:16
```

Die einzelnen Werte werden im Glossar (Abschnitt 1.3) erklärt. Die Reihenfolge und Formatierung muss beibehalten werden. Die Umgebungsvariable wird nur verwendet, wenn die Software (oder das Migrationswerkzeug) neue Überprüfungscodes erstellt, beispielsweise wenn der Benutzer ein neues Passwort setzt. Sonst werden die gespeicherten Werte aus der Konfiguration verwendet.

`srp6.disable.login`: Deaktiviert die neue Authentifizierung mit SRP und versucht in jedem Fall die alte Authentifizierung ohne Verschlüsselung. Nur für Testzwecke bzw. zur Problemdiagnose.

`srp6.disable.useradmin`: Deaktiviert die neue Benutzerverwaltungsschnittstelle und verwendet die alte `UserAdministration`-Schnittstelle. Nur für Testzwecke bzw. zur Problemdiagnose.

`srp6.disable.verifier`: Sorgt dafür, dass beim Setzen von Passwörtern über die Benutzerverwaltungsfunktionen Klartextpasswörter statt Überprüfungscodes an die Konfiguration übertragen werden. Dies ist notwendig, wenn sich noch alte Softwareversionen einloggen können müssen. Die Daten werden in jedem Fall verschlüsselt übertragen.

5.1.2 Aufrufparameter

```
-verschluesselung=[immer,automatisch,nein]
```

```
-erlaubeHmacAuthentifizierung=[ja,nein]
```

5.2 Mögliche Probleme

Im folgenden Abschnitt werden Lösungshinweise für mögliche Probleme aufgelistet.

Der Datenverteiler startet nicht mehr oder beendet sich kurz nach dem Start: Datenverteiler-Debug-Ausgaben nach einer der folgenden Debug-Meldungen durchsuchen. Wenn sich der Datenverteiler nicht bei der Konfiguration authentifizieren kann, sicherstellen, dass der Datenverteilerbenutzer in der benutzerverwaltung.xml vorhanden ist, dort ein Admin ist, und ein entsprechendes Systemobjekt für den Benutzer existiert.

Interaktiver Login beim Datenverteiler schlägt fehl: Bei Verwendung von Einmalpasswörtern den Passwortindex an den Benutzernamen anhängen (z. B. “Tester–13”). Mit dem Migrationstool das gespeicherte Passwort überprüfen (”Passwort überprüfen). Notfalls ein neues Passwort setzen. Wenn das Passwort verschlüsselt gespeichert ist, aber eine alte Kernsoftware eingesetzt wird, entweder Kernsoftware aktualisieren oder das Passwort wieder im Klartext speichern (durch Bearbeitung von der `benutzerverwaltung.xml` bei beendeter Konfiguration oder durch das Verwenden der Benutzerverwaltungsfunktionen (ggf. über das Migrationswerkzeug) mit gesetzter Umgebungsvariable (Unterabschnitt 5.1.1) `srp6.disable.verifier`).

Automatischer Login beim Datenverteiler schlägt fehl: Ein Login-Token wird bei jeder Passwortänderung ungültig, auch wenn das Passwort beibehalten wurde oder wieder zurück geändert wurde. Daher muss in dem Fall ein neuer Login-Token erzeugt werden. Ansonsten gelten die gleichen Hinweise wie beim interaktiven Login.

Performanceprobleme: Falls der Login zu lange dauert, gibt es folgende Möglichkeiten, die Software zu beschleunigen

- Sicherstellen, dass niemand Brute-Force-Angriffe durchführt, also beim Einloggen sehr viele Passwörter durchprobiert. Datenverteiler und Konfiguration bremsen in solchen Fällen weitere Anmeldeversuche aus Sicherheitsgründen aus.
- Kryptographischen Parameter **c** reduzieren (verringert aber die Passwortsicherheit, insbesondere bei schwachen Passwörtern).

Falls der Datenverteiler im Betrieb zu langsam ist kann die Verschlüsselung für einzelne Applikationen deaktiviert werden (Abschnitt 3.3).

5.2.1 Debug-Ausgaben

Die Software erzeugt folgende neue Debug-Meldungen, die sich auf die neue Authentifizierung und Verschlüsselung beziehen. Warnungen behindern normalerweise nicht den Betrieb, können aber auf Sicherheitsprobleme oder Konfigurationsfehler hindeuten und sollten möglichst schnell behoben werden. Ausgaben mit Level “Fehler” stellen ein ernstes Problem dar und müssen behoben werden damit die Software korrekt funktioniert.

Tabelle 5.1: Debug-Ausgaben mit Level *Warnung*

Meldung	Erklärung
Verschlüsselte Anmeldung per SRP fehlgeschlagen, da die Konfiguration keine Verschlüsselung unterstützt	Datenverteiler und Applikation verwenden bereits die neue Authentifizierung, aber die Konfiguration wurde noch nicht aktualisiert, sodass der Datenverteiler diese nicht nach den Überprüfungscodes fragen kann. Es wird die alte Authentifizierung ohne Verschlüsselung verwendet.
Der Datenverteiler unterstützt keine Verschlüsselung und sollte aktualisiert werden	Der Datenverteiler wurde noch nicht auf die neue Version aktualisiert. Es wird die alte Authentifizierung ohne Verschlüsselung verwendet.
Kann Loopback-Status der Verbindung nicht bestimmen	Es kann nicht festgestellt werden, ob Datenverteiler und Applikation bzw. anderer Datenverteiler auf dem selben Rechner laufen oder nicht. Bei Verwendung des Arguments `-verschluesselung=auto` wird daher davon ausgegangen, dass beide sich auf unterschiedlichen Rechnern befinden und es wird im Zweifelsfall verschlüsselt.
Benutzer … möchte sich über die alte Hmac-Authentifizierung authentifizieren, aber das ist aus Sicherheitsgründen nicht erlaubt. (Aufrufparameter: -erlaubeHmacAuthentifizierung)	Ausgabe am Datenverteiler, wenn eine Applikation sich über das alte Verfahren authentifizieren will, das aber nicht erlaubt wurde.
Angegebener Passwort-Index ist nicht am Benutzer … vorhanden: …	Ausgabe der Konfiguration wenn der Login mit einem Einmalpasswort fehlschlägt, weil das Einmalpasswort mit dem angegebenen Index nicht vorhanden ist oder bereits benutzt wurde.
Ungültige Telegrammabfolge: …	Der Client versendet die Telegramme zur Authentifizierung in der falschen Reihenfolge und hält das Protokoll nicht ein.
Benutzer ist nicht authentifiziert	Der Empfang von verschiedenen Telegrammen (z. B. Nutzdaten) ist nicht möglich, während der Client noch nicht eingeloggt ist.
Benutzer ist bereits authentifiziert	Der Client startet einen neuen Login-Versuch, obwohl er bereits korrekt authentifiziert ist.

Tabelle 5.2: Exceptions und Debug-Ausgaben mit Level *Fehler*

Meldung	Erklärung
Fehler beim Deaktivieren eines Einmal-Passworts: …	Ein Einmalpasswort kann nicht auf ungültig gesetzt werden. Möglicherweise ist die benutzerverwaltung.xml nicht beschreibbar. Das kann dazu führen, dass ein Einmalpasswort mehrmals zur Authentifizierung benutzt werden kann.
Unverschlüsseltes Telegramm bei bestehender Verschlüsselung empfangen: …	Es wurde ein unverschlüsseltes Telegramm empfangen obwohl eine verschlüsselte Verbindung besteht. Möglicherweise handelt es sich um einen Softwarefehler beim verbundenen System oder ein Angreifer manipuliert die Verbindung. Die Verbindung wird terminiert.
Verschlüsseltes Telegramm erhalten, aber keine Verschlüsselung aktiv	Es wurde ein verschlüsseltes Telegramm empfangen obwohl keine Verschlüsselung aktiv ist. Das Telegramm kann nicht entschlüsselt werden, die Verbindung wird terminiert.
Ungültige kryptographische Parameter	Fehler beim Empfang von kryptographischen Parametern von der Konfiguration.
Der Datenverteiler antwortet nicht bei der SRP-Authentifizierung	Der Datenverteiler reagiert nicht auf die Authentifizierungsanfrage. Möglicherweise antwortet die Konfiguration nicht. Debug-Ausgaben des Datenverteilers prüfen.
Datenverteiler und/oder Konfiguration sind veraltet und unterstützen die sichere SRP-Authentifizierung nicht, und die alte Authentifizierung ist nicht erlaubt.	Die alte Authentifizierung wurde deaktiviert und die neue kann aufgrund von veralteter Software nicht verwendet werden.
Leeres Passwort	Passwörter mit Länge 0 werden nicht mehr zugelassen.
Der Datenverteiler konnte sich nicht mit dem Benutzer "…" bei der Konfiguration authentifizieren	Der Benutzername, unter dem der Datenverteiler gestartet wird (Aufrufparameter `-benutzer=`) ist in der Konfiguration nicht vorhanden, oder das Passwort/der Login-Token in der Authentifizierungsdatei des Datenverteilers passt nicht zum Passwort/Überprüfungscode in der benutzerverwaltung.xml der Konfiguration.

5.3 Technische Details zur Verschlüsselung

5.3.1 Authentifizierung

- Authentifizierungsverfahren: SRP6a (Implementierung: NimbusSRP)
- X-Routine: `H(s | H(u | ":" | KDF(p, s)))`
- Hashfunktion `H`: Standardmäßig `SHA-256`
- Schlüsselableitungsfunktion `KDF`: Standardmäßig `PBKDF2WithHmacSHA256` mit 20.000 Iterationen und 256 Bit Rückgabe
- SRP-Primzahl `N`: Standardmäßig 1024 Bit aus RFC 5054[1]

5.3.2 Verschlüsselung

- Verschlüsselungsverfahren: AES-GCM
- Schlüssellänge: Standardmäßig 128 Bit
- GCM-Tag-Bits: Standardmäßig 96 Bit
- Ableitung der AES-Schlüssel aus dem gemeinsamen SRP-Sitzungsschlüssel `S` (bei AES–128):

  `S* = H(0x00000000 | S) | H(0x00000001 | S) | H(0x00000002 | S)` …

  = Schlüssel(ClientServer) | Schlüssel(ServerClient) | Nonce-Salt(ClientServer) | Nonce-Salt(ServerClient)

  (| = Verkettung)

  Nonce-Berechnung aus RFC 5288[2] mit Verwendung von einem Telegrammzähler als explizitem Nonce-Teil.

[1] https://tools.ietf.org/html/rfc5054#appendix-A
[2] https://tools.ietf.org/html/rfc5288#section-3