

# Dm-Verity 适配指导手册

文档版本            V2.1  
发布日期            2020-08-18

**版权所有 © 紫光展锐科技有限公司。保留一切权利。**

本文件所含数据和信息都属于紫光展锐所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负责任何与本文件相关的直接或间接的、任何伤害或损失。

请参照交付物中说明文档对紫光展锐交付物进行使用，任何人对紫光展锐交付物的修改、定制化或违反说明文档的指引对紫光展锐交付物进行使用造成的任何损失由其自行承担。紫光展锐交付物中的性能指标、测试结果和参数等，均为在紫光展锐内部研发和测试系统中获得的，仅供参考，若任何人需要对交付物进行商用或量产，需要结合自身的软硬件测试环境进行全面的测试和调试。非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

# 紫光展锐科技有限公司



# 前言

## 概述

本文档主要介绍了 Android Verified Boot (AVB) 过程中对于镜像较大分区（比如 **system** 分区）的校验原理以及实现方式。

## 读者对象


Android Verified Boot 和 Dm-Verity 相关研发及测试人员。

## 缩略语

缩略语	英文全名	中文解释
AVB	Android Verified Boot	Android 启动时验证

## 符号约定

在本文中可能出现下列标志，它所代表的含义如下。

符号	说明
 说明	用于突出重要/关键信息、补充信息和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害。

## 变更信息

文档版本	发布日期	修改说明
V1.0	2015-02-20	第一次正式发布。
V2.0	2019-11-08	增加 Android 10.0 平台适配
V2.1	2020-08-18	<ul style="list-style-type: none"><li>增加 Android 11.0 平台适配</li><li>更新模板</li></ul>

文档版本	发布日期	修改说明
		<ul style="list-style-type: none"><li>文档名由《Android Dm-Verity 适配指导文档》改为《Dm-Verity 适配指导手册》</li></ul>

## 关键字

Dm-Verity、AVB、Verified Boot。

# 目 录

1 Dm-Verity 概述 .....	1
1.1 Android Verified Boot.....	1
1.2 Dm-Verity 实现.....	1
2 Dm-Verity 适配 .....	3
2.1 Android 6.0 平台适配.....	3
2.1.1 功能适配.....	3
2.1.2 定制 key.....	6
2.1.3 功能验证.....	7
2.1.4 功能关闭.....	7
2.2 Android 7.0 平台适配.....	7
2.2.1 功能适配.....	8
2.2.2 定制 key.....	8
2.2.3 功能验证.....	8
2.2.4 功能关闭.....	9
2.3 Android 8.1 平台适配.....	9
2.3.1 功能适配.....	9
2.3.2 定制 key.....	10
2.3.3 功能验证.....	11
2.3.4 功能关闭.....	11
2.4 Android 9.0 平台适配.....	11
2.4.1 功能适配.....	12
2.4.2 定制 key.....	14
2.4.3 功能验证.....	14
2.4.4 功能关闭.....	15
2.5 Android 10.0 平台适配.....	15
2.5.1 功能适配.....	16
2.5.2 定制 key.....	17
2.5.3 功能验证.....	18
2.5.4 功能关闭.....	19
2.6 Android 11.0 平台适配.....	19
2.6.1 功能适配.....	20
2.6.2 定制 key.....	21
2.6.3 功能验证.....	21
2.6.4 功能关闭.....	23
3 常见问题.....	24

---

3.1 性能影响 .....	24
3.2 OTA 影响 .....	24
3.3 Android 9.0 userdebug 版本 remount 失败.....	24
3.4 功能异常排查指引 .....	25
4 参考文档 .....	26

## 图目录

---

图 1-1 哈希树结构图 .....	2
--------------------	---

# 1 Dm-Verity 概述

## 1.1 Android Verified Boot

对于要启动的 Android 版本中包含的所有可执行代码和数据，启动前均要求以加密形式对其进行验证，包括内核（从 **boot** 分区加载）、设备树（从 **dtbo** 分区加载）、**system** 分区和 **vendor** 分区等。

仅读取一次的小分区（例如 **boot** 和 **dtbo**），通常是通过将整个内容加载到内存中，然后计算相应哈希值来进行验证。接下来，系统会比较这个计算出的哈希值与预期哈希值。如果值不一致，则 Android 将无法加载。

内存装不下的较大分区（例如 **system** 和 **vendor**），可以使用哈希树进行验证。此时，验证流程会在将数据加载到内存时持续进行。在这种情况下，系统会在运行时计算哈希树的根哈希值，并对照预期根哈希值进行检查。Android 包含用于验证较大分区的 Dm-Verity 内核驱动程序。如果在某个时间点计算出的根哈希值与预期根哈希值不一致，系统便不会使用相应数据，而且 Android 会出现错误。

预期哈希值通常存储在每个已验证分区的末尾或开头、专用分区中，或同时位于以上两个位置。最重要的是，这些哈希值是由信任根以直接或间接的方式签名的。

## 1.2 Dm-Verity 实现

Dm-Verity 是一项内核功能，Android 从 5.1 版本开始支持。它可以提供透明的对块设备的完整性校验，以防止恶意程序对系统分区的篡改。

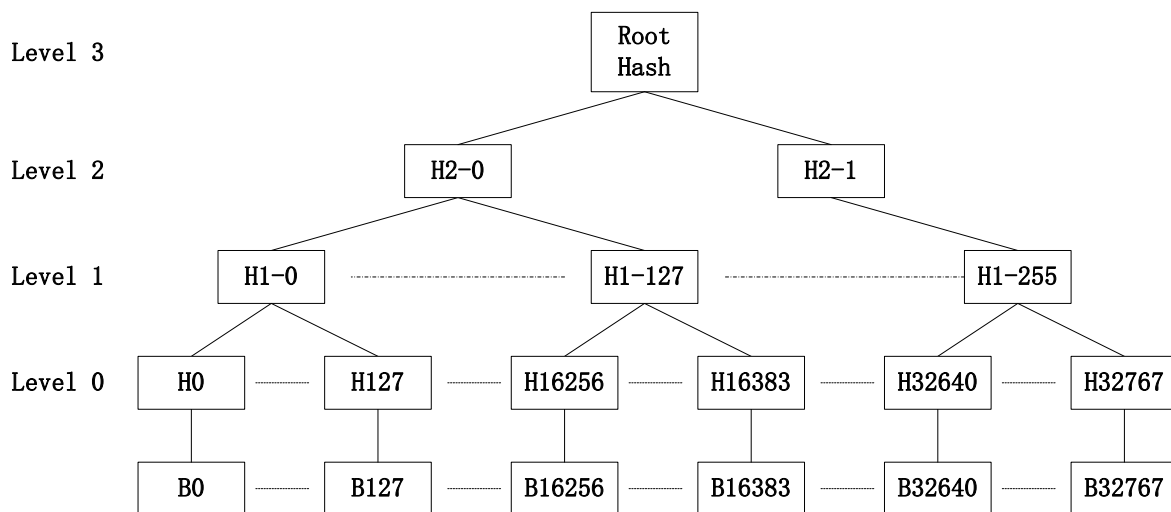
Dm-Verity 虚拟了一个块设备，当对块数据进行读取时会首先进行哈希计算，并与预先计算好的哈希树进行校验，如果匹配则读取成功，否则会生成一个读取的 I/O 错误。以此来达到对数据进行完整性校验的目的。

这棵预先计算好的哈希树包含了要校验的目标设备的所有块，对于每个块（一般是 4KB），有一个 SHA 散列（32B），它存储在树的叶子节点。树的中间节点则是对这些叶节点的再次 SHA 散列（还是以 4KB 为一个单位），通过这样多次反复的哈希计算，直到得出唯一的哈希值为止。这个唯一的数值称为 Root Hash。那么基于散列的特性，当任意一个 block 有任意的变化，都会导致 Root Hash 数值发生变化。

以一个包含 32768 个块的设备为例，哈希树的结构图如下：



图1-1 哈希树结构图



# 2 Dm-Verity 适配

## 2.1 Android 6.0 平台适配

### 2.1.1 功能适配

#### 配置 kernel config

通过 `kuconfig` 命令配置 kernel config `DM_VERITY`，配置路径如下：

```
.config - Linux/arm 3.10.65 Kernel Configuration
> Search (dm_verity)

Symbol: DM_VERITY [=y]
Type : tristate
Prompt: Verity target support
Location:
  -> Device Drivers
(1) -> Multiple devices driver support (RAID and LVM) (MD [=y])
Defined at drivers/md/Kconfig:399
Depends on: MD [=y] && BLK_DEV_DM [=y]
Selects: CRYPTO [=y] && CRYPTO_HASH [=y] && DM_BUFIO [=y]
```

修改生效的文件是 `/kernel/arch/arm/arm64/configs/` 下面对应工程的配置文件：

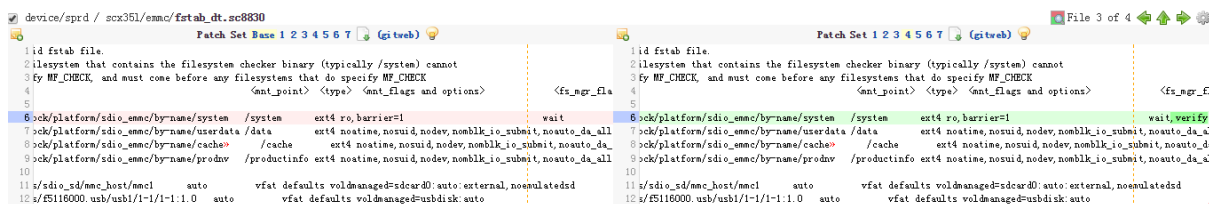
# CONFIG_DM_FLAKEKEY is not set	1254 # CONFIG_DM_FLAKEKEY is not set
# CONFIG_DM_VERITY is not set	1255 CONFIG_DM_BUFIO=y
	1256 CONFIG_DM_VERITY=y
# CONFIG_TARGET_CORE is not set	1257 # CONFIG_TARGET_CORE is not set

#### 修改 Board 相关配置

步骤 1 在 `/dev/sprd/scx35/common/device.mk`，增加 Dm-Verity 的编译配置，同时用 `TARGET_DM_VERITY` 宏控制。

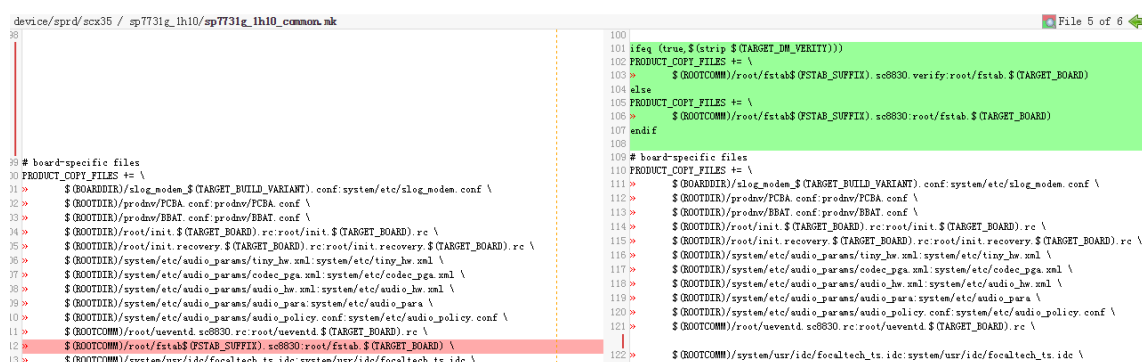
```
device/sprd/scx35 / common/device.mk
Patch Set Base 1 2 (gitweb)
... skipped 238 common lines ...+10.0
239 else
240 ifeq ($(strip $(PRODUCT_RAM)),high)
241 $(call inherit-product, frameworks/native/build/phone-xhdpi-1024-dalvik-heap.mk)
242 PRODUCT_PROPERTY_OVERRIDES += \
243 > ro.board_ram_size=high \
244 > ro.config.low_ram=false \
245 > ro.product.ram=high
246 endif
247 endif
248 # 0}
249
249 ifeq (true,$(strip $(TARGET_DM_VERITY)))
250 $(call inherit-product, build/target/product/verity.mk)
251 PRODUCT_SYSTEM_VERITY_PARTITION := /dev/block/platform/sdio_ennn/by-name/system
252 endif
```

## 步骤 2 fstab 中对 system 分区增加校验标志。

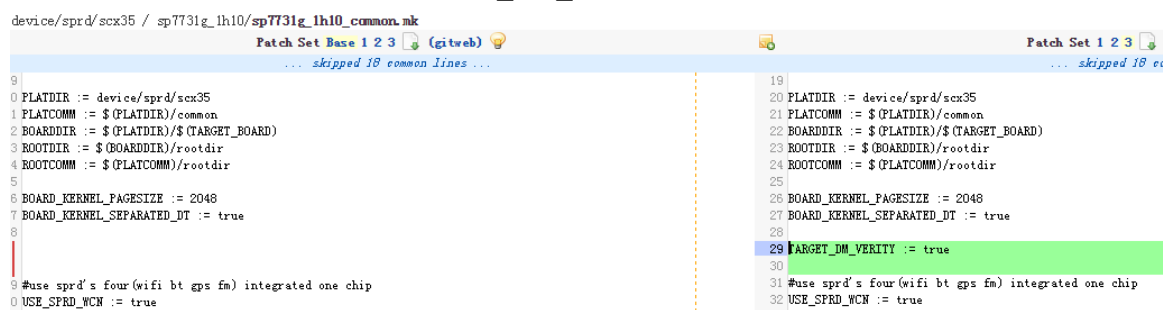


如果直接修改，可能导致兼容性问题，建议用以下方式进行修改：

- 增加一个 fstab 文件：fstab.sc8830.verify。这个文件拷贝自 fstab.sc8830，并参考上图所展示的差异进行修改。
- 在对应的 Board 中通过 TARGET\_DM\_VERITY 宏来控制拷贝的 fstab 文件。参考下图进行修改。



- 在需要打开此功能的 Board 将 TARGET\_DM\_VERITY 置为 true。参考下图进行修改。



## ---结束

## 修改签名程序及编译脚本

UNISOC 的 kernel 中增加了 device tree 功能，因此需要修改对应的签名程序以及编译脚本，增加 dt 的相关信息，如下所示。

```
platform/system/extras / verity/BootSignature.java
Patch Set Base 1 2 (gitweb)
... skipped 195 common lines ...+105
196
197 image.getLong(); // magic
198 int kernelSize = image.getInt0;
199 image.getInt0; // kernel_addr
200 int randiskSize = image.getInt0;
201 image.getInt0; // randisk_addr
202 int secondSize = image.getInt0;
203 image.getLong(); // second_addr + tags_addr
204 int pageSize = image.getInt0;
205
206
207 int length = pageSize // include the page aligned image header
208 + (kernelSize + pageSize - 1) / pageSize * pageSize
209 + (randiskSize + pageSize - 1) / pageSize * pageSize
210 + ((secondSize + pageSize - 1) / pageSize * pageSize;
211
212 length = (length + pageSize - 1) / pageSize * pageSize;
213
214
215
Patch Set 1 2 (gitweb)
... skipped 195 common lines ...+105
196
197 image.getLong(); // magic
198 int kernelSize = image.getInt0;
199 image.getInt0; // kernel_addr
200 int randiskSize = image.getInt0;
201 image.getInt0; // randisk_addr
202 int secondSize = image.getInt0;
203 image.getLong(); // second_addr + tags_addr
204 int pageSize = image.getInt0;
205
206
207 int dt_Size = image.getInt0;
208
209 int length = pageSize // include the page aligned image header
210 + (kernelSize + pageSize - 1) / pageSize * pageSize
211 + (randiskSize + pageSize - 1) / pageSize * pageSize
212 + ((secondSize + pageSize - 1) / pageSize * pageSize
213 + ((dt_Size + pageSize - 1) / pageSize * pageSize;
214
215 length = (length + pageSize - 1) / pageSize * pageSize;
216
217
218
platform/build / core/Makefile
Patch Set Base 1 2 3 4 5 6 7 (gitweb)
... skipped 521 common lines ...+105
522 $(call pretty,"Target boot image: $@")
523 $(hide) $(MKBOOTIMG) $(INTERNAL_BOOTIMAGE_ARGS) --output $@
524
525 bootimage-nodups
526 $(MKBOOTIMG)
527 @echo "make $@: ignoring dependencies"
528 $(hide) $(MKBOOTIMG) $(INTERNAL_BOOTIMAGE_ARGS) --output $(INSTALLED_BOOTIMAGE_TARGET)
529
530 seq (true,$(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_SUPPORTS_VERITY)) # TARGET_BOOTIMAGE_USE_EXT2 != true
531
532 $(INSTALLED_BOOTIMAGE_TARGET): $(MKBOOTIMG) $(INTERNAL_BOOTIMAGE_FILES) $(BOOT_SIGNER)
533 $(call pretty,"Target boot image: $@")
534 $(hide) $(MKBOOTIMG) $(INTERNAL_BOOTIMAGE_ARGS) $(BOARD_MKBOOTIMG_ARGS) --output $@
535 $(BOOT_SIGNER) /boot $@ $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_VERITY_SIGNING_KEY).pk8 $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_VERITY_SIGNING_KEY).pk8 $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_VERITY_SIGNING_KEY).pk8 $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_VERITY_SIGNING_KEY).pk8
536 $(hide) $(call assert-max-image-size,$@,$(BOARD_BOOTIMAGE_PARTITION_SIZE))
537
538 bootimage-nodups
539 $(MKBOOTIMG) $(BOOT_SIGNER)
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
```

完成上述配置后，需要做一次完整编译，并重新烧录 boot.img、system.img、userdata.img，即可在 userdebug 版本关闭 Dm-Verity 功能。

## 2.1.2 定制 key

### Key 文件作用说明

Dm-Verity 包含三个 key 文件，路径位于：build/target/product/security/，具体作用为：

- verity.pk8：这是一个私钥，用于给 boot.img 和 system.img 签名。
- verity.x509.pem：这是一个证书，此证书内包含有公钥信息。
- verity\_key：这是一个公钥，用于 system.img 的 Dm-Verity 完整性校验。

### 替换原生 key

替换步骤如下：

步骤 1 生成 verity 相关的三个 key 文件：verity.pk8、verity.x509.pem、verity\_key：

- 在 Linux 系统中，确保所安装 openssl 版本在 1.0 以上（ubuntu 终端输入 “openssl version” 查看版本号，比如：OpenSSL 1.0.1f 6 Jan 2014 是 OK 的）。
- 终端切换到 IDH 代码的根目录下，输入如下命令（直接回车，不用输入密码），就会在当前根目录生成 verity.pk8 和 verity.x509.pem：

```
development/tools/make_key verity '/C=US/ST=California/L=Mountain View/O=Android/OU=Android/CN=Android/emailAddress=android@android.com'
```

- 在终端中执行 source build/envsetup.sh、lunch 选择对应的工程、kheader 后，再输入 make generate\_verity\_key 或者 mmm system/extras/verity/，就会生成 generate\_verity\_key。
- 在终端继续输入如下命令，将会在 idh.code 根目录生成 verity\_key.pub。

```
out/host/linux-x86/bin/generate_verity_key -convert verity.x509.pem verity_key
```

- 将 verity\_key.pub 重命名为 verity\_key。

步骤 2 将上述生成的三个 key 替换到 build/target/product/security/目录下。

步骤 3 重新进行完整版本编译。

----结束

验证 key 是否替换成功：

步骤 1 verity\_key 对比验证

verity\_key 最后会被打包进 boot.img 中。以 SC9832E 为例，编译时生成的目录位于：out/target/product/sp9832e\_1h10/root。对比上述目录下的 verity\_key 和前述用命令生成的 verity\_key，如果两者一致则说明替换成功。

步骤 2 交叉验证

基于同一软件版本，分别使用 A 和 B 两组不同的 key 编译得到 A 和 B 两个完整版本（确保仅 verity 的 key 不同）。验证步骤如下：

- 下载版本 A 至手机中，开机并按照“验证 Dm-Verity 功能”说明，确认 Dm-Verity 功能开启正常。预期结果：手机开机正常，并且 system 分区按照 dm-0 的方式挂载。
- 下载替换版本 B 的 system.img，开机验证。预期结果：手机无法开机。
- 继续下载替换版本 B 的 boot.img，开机验证。预期结果：手机开机正常，system 分区按照 dm-0 的方式挂载。

----结束

## 2.1.3 功能验证

验证此功能需要在 user 版本上验证，因为 userdebug 版本上该功能是关闭的。

验证功能是否成功开启的办法：查看 system 分区的挂载方式。当挂载方式为 dm-x 时，表明功能开启成功，如下图所示。反之则是未开启。

```
adb shell mount
rootfs / rootfs ro,seclabel,size=369544k,nr_inodes=92386 0 0
tmpfs /dev tmpfs rw,seclabel,nosuid,relatime,nodev=755 0 0
devpts /dev/pts devpts rw,seclabel,relatime,nodev=600 0 0
proc /proc proc rw,relatime 0 0
sysfs /sys sysfs rw,seclabel,relatime 0 0
selinuxfs /sys/fs/selinux selinuxfs rw,relatime 0 0
none /acct cgroup rw,relatime,cpusacct 0 0
none /sys/fs/cgroup tmpfs rw,seclabel,relatime,nodev=750,gid=1000 0 0
tmpfs /mnt tmpfs rw,seclabel,relatime,nodev=755,gid=1000 0 0
none /dev/cpuctl cgroup rw,relatime,cpu 0 0
/dev/block/dm-0 /system ext4 ro,seclabel,relatime,data=ordered 0 0
/dev/block/platform/soc/soc:ap-ahb/50430000.sdio/by-name/userdata /data ext4 rw,seclabel,nosuid,nodev,noatime,noauto_da_alloc,data=ordered 0
/dev/block/platform/soc/soc:ap-ahb/50430000.sdio/by-name/cache /cache ext4 rw,seclabel,nosuid,nodev,noatime,noauto_da_alloc,data=ordered 0
/dev/block/platform/soc/soc:ap-ahb/50430000.sdio/by-name/prodinfo /productinfo ext4 rw,seclabel,nosuid,nodev,noatime,noauto_da_alloc,data=ordered 0
adb /dev/usb-lun0 adb functionfs rw,relatime 0 0
tmpfs /storage tmpfs rw,seclabel,relatime,nodev=755,gid=1000 0 0
/dev/fuse /mnt/runtime/default/emulated fuse rw,nosuid,nodev,noexec,noatime,user_id=1023,group_id=1023,default_permissions,allow_other 0 0
/dev/fuse /storage/emulated fuse rw,nosuid,nodev,noexec,noatime,user_id=1023,group_id=1023,default_permissions,allow_other 0 0
/dev/fuse /mnt/runtime/read/emulated fuse rw,nosuid,nodev,noexec,noatime,user_id=1023,group_id=1023,default_permissions,allow_other 0 0
/dev/fuse /mnt/runtime/write/emulated fuse rw,nosuid,nodev,noexec,noatime,user_id=1023,group_id=1023,default_permissions,allow_other 0 0
```

## 2.1.4 功能关闭

将 TARGET\_DM\_VERITY 置为 false，即可关闭。

## 2.2 Android 7.0 平台适配

Android 7.0 中针对 Dm-Verity 增加了一项新的子项功能：DM\_VERITY\_FEC，即 Verity forward error correction。由于此功能的引入，导致 system.img 格式发生变化，校验方式也跟着发生变化。这就要求 system.img 的大小和 xml 中定义的 system 分区大小保持一致。否则系统无法找到校验数据，会导致 system 分区无法挂载，最终无法开机。

为了满足上述新功能需求，平台做了以下功能性修改：

- 增加 system 分区自适应功能。通过 SYSTEM\_IMAGE\_SIZE\_ADAPT 宏来控制。
  - 当 SYSTEM\_IMAGE\_SIZE\_ADAPT 为 false 的时候，功能关闭。
  - 当 SYSTEM\_IMAGE\_SIZE\_ADAPT 为 true 的时候，功能开启。
- 针对 system 分区特殊大小（比如 1600M），进行分区大小再调整的修改。具体方案为：当 system 分区自适应功能打开时，在编译脚本中自动进行分区调整。当 system 分区自适应功能关闭时，在编译阶段报错，并主动提示用户需要配置 BOARD\_SYSTEMIMAGE\_PARTITION\_SIZE 的大小。

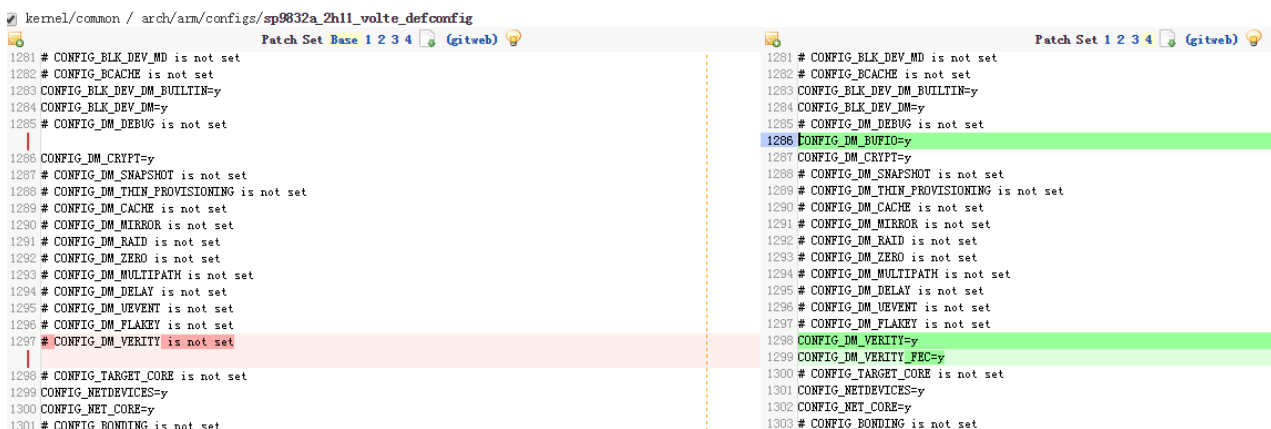
## 说明

- 当 system 分区为 1600M 的时候，无法分配出合适的 system 文件系统（以 A 表示）和校验数据的大小（以 B 表示），因为 B 的大小是跟随 A 的大小而变化的。比如，当 A 为 1651507200 时，B 计算出的大小为 26210304，总大小为 1677717504（刚好比 1600M 少一个 block，也就是 4096）。当 A 为 1651511296，B 计算出的大小为 26214400，总大小为 1677725696（刚好比 1600M 多一个 block，也就是 4096）。其中 B 包含三部分：B1（GetVerityTreeSize）、B2（GetVerityMetadataSize）、B3（GetVerityFECSize）。
- 文件系统的大小规则：假设传进来的镜像大小是 X，X 与 128M 求余，余数为 Y。如果  $0 < Y < 4M$ ，则文件系统会把 Y 丢掉。如果大于等于 4M，则文件系统不会丢 block。

平台默认开启 system 分区大小自适应功能。如需关闭该功能，则对 system 分区的大小进行调整时，需要同时修改 xml 中的 system 分区大小和 BOARD\_SYSTEMIMAGE\_PARTITION\_SIZE，使两者保持大小一致。同时，配置分区大小的数值需要为 50M 的倍数，并且排除 1600M 这个特殊值。

## 2.2.1 功能适配

### 配置 kernel config



其余配置请参考“2.1.1 功能适配”。

## 2.2.2 定制 key

Android 7.0 定制 key 的方式与 Android 6.0 相同，请参考“2.1.2 定制 key”。

## 2.2.3 功能验证

验证此功能需要在 user 版本上验证，因为 userdebug 版本上该功能是关闭的。

验证功能是否成功开启的办法：查看 system 分区的挂载方式。当挂载方式为 dm-x 时，表明功能开启成功，如下图所示。反之则是未开启。



```
adb shell mount
rootfs / rootfs ro,seclabel,size=369544k,nr_inodes=92386 0 0
tmpfs /dev tmpfs rw,seclabel,nosuid,relatime,nodev=755 0 0
devpts /dev/pts devpts rw,seclabel,relatime,nodev=600 0 0
proc /proc proc rw,relatime 0 0
sysfs /sys sysfs rw,seclabel,relatime 0 0
selinuxfs /sys/fs/selinux selinuxfs rw,relatime 0 0
none /acct cgroup rw,relatime,cpuacct 0 0
none /sys/fs/cgroup tmpfs rw,seclabel,relatime,nodev=755,gid=1000 0 0
tmpfs /mnt tmpfs rw,seclabel,relatime,nodev=755,gid=1000 0 0
none /dev/cpuctl cgroup rw,relatime,cpu 0 0
/dev/block/dm-0 /system ext4 ro,seclabel,relatime,data=ordered 0 0
/dev/block/platform/soc/soc:ap-ahb/50430000.sdio/by-name/userdata /data ext4 rw,seclabel,nosuid,nodev,noatime,noauto_da_alloc,data=ordered 0
/dev/block/platform/soc/soc:ap-ahb/50430000.sdio/by-name/cache /cache ext4 rw,seclabel,nosuid,nodev,noatime,noauto_da_alloc,data=ordered 0
/dev/block/platform/soc/soc:ap-ahb/50430000.sdio/by-name/prodinfo /productinfo ext4 rw,seclabel,nosuid,nodev,noatime,noauto_da_alloc,data=ordered 0
adb /dev/usb-ffs/adb functionfs rw,relatime 0 0
tmpfs /storage tmpfs rw,seclabel,relatime,nodev=755,gid=1000 0 0
/dev/fuse /mnt/runtime/default/emulated fuse rw,nosuid,nodev,noexec,noatime,user_id=1023,group_id=1023,default_permissions,allow_other 0 0
/dev/fuse /storage/emulated fuse rw,nosuid,nodev,noexec,noatime,user_id=1023,group_id=1023,default_permissions,allow_other 0 0
/dev/fuse /mnt/runtime/read/emulated fuse rw,nosuid,nodev,noexec,noatime,user_id=1023,group_id=1023,default_permissions,allow_other 0 0
/dev/fuse /mnt/runtime/write/emulated fuse rw,nosuid,nodev,noexec,noatime,user_id=1023,group_id=1023,default_permissions,allow_other 0 0
```

## 2.2.4 功能关闭

将 TARGET\_DM\_VERITY 置为 false，即可关闭。

## 2.3 Android 8.1 平台适配

在 Android 8.1 上适配 Dm-Verity 功能与之前的 Android 版本有所不同。

- Verified boot 在 Android 8.1 采用了 AVB2.0 方案。在 Android 6.0 和 Android 7.0 版本均采用 AVB1.0 方案。因此 Dm-Verity 作为 verified boot 功能的一部分，也采用了 AVB2.0 方案。
- Android 8.1 上 system 和 vendor 分区的挂载逻辑提前到 DoFirstStageMount() 函数中完成，故校验 flag 的配置需要在 dts 文件中定义。

### 2.3.1 功能适配

#### 配置 kernel config

```
kernel/common / arch/arm/configs/sp9832a_2h1l_volte_defconfig
Patch Set Base 1 2 3 4 (gitweb)
1281 # CONFIG_BLK_DEV_MD is not set
1282 # CONFIG_BCACHE is not set
1283 CONFIG_BLK_DEV_DM_BUILTIN=y
1284 CONFIG_BLK_DEV_DM=y
1285 # CONFIG_DM_DEBUG is not set
1286 CONFIG_DM_CRYPT=y
1287 # CONFIG_DM_SNAPSHOT is not set
1288 # CONFIG_DM_THIN_PROVISIONING is not set
1289 # CONFIG_DM_CACHE is not set
1290 # CONFIG_DM_MIRROR is not set
1291 # CONFIG_DM_RAID is not set
1292 # CONFIG_DM_ZERO is not set
1293 # CONFIG_DM_MULTIPATH is not set
1294 # CONFIG_DM_DELAY is not set
1295 # CONFIG_DM_UEVENT is not set
1296 # CONFIG_DM_FLAKE is not set
1297 # CONFIG_DM_VERITY is not set
1298 # CONFIG_TARGET_CORE is not set
1299 CONFIG_NETDEVICES=y
1300 CONFIG_NET_CORE=y
1301 # CONFIG_BONDING is not set
Patch Set 1 2 3 4 (gitweb)
1281 # CONFIG_BLK_DEV_MD is not set
1282 # CONFIG_BCACHE is not set
1283 CONFIG_BLK_DEV_DM_BUILTIN=y
1284 CONFIG_BLK_DEV_DM=y
1285 # CONFIG_DM_DEBUG is not set
1286 CONFIG_DM_CRYPT=y
1287 # CONFIG_DM_SNAPSHOT is not set
1288 # CONFIG_DM_THIN_PROVISIONING is not set
1289 # CONFIG_DM_CACHE is not set
1290 # CONFIG_DM_MIRROR is not set
1291 # CONFIG_DM_RAID is not set
1292 # CONFIG_DM_ZERO is not set
1293 # CONFIG_DM_MULTIPATH is not set
1294 # CONFIG_DM_DELAY is not set
1295 # CONFIG_DM_UEVENT is not set
1296 # CONFIG_DM_FLAKE is not set
1297 # CONFIG_DM_FLAKE is not set
1298 CONFIG_DM_VERITY=y
1299 CONFIG_DM_VERITY_FEC=y
1300 # CONFIG_TARGET_CORE is not set
1301 CONFIG_NETDEVICES=y
1302 CONFIG_NET_CORE=y
1303 # CONFIG_BONDING is not set
```

#### 修改 dts 文件，开启 Dm-Verity flag

适配 system 和 vendor 分区 Dm-Verity 挂载方式，需要修改 dts 文件。在 dts 文件中添加 avb flag，修改方案有两种：

- 方案一：system 和 vendor 的 fsmgr\_flags 从 “wait” 改成 “wait,avb”，修改文件 kernel/arch/arm/boot/dts/sp9832e-common.dtsi:



```

firmware {
    android {
        compatible = "android,firmware";

        vbmeta {
            compatible = "android,vbmeta";
            parts = "vbmeta,boot,recovery,system,vendor";
        };

        fstab {
            compatible = "android,fstab";
            fs_system: system {
                compatible = "android,system";
                dev = "/dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/system";
                type = "ext4";
                mnt_flags = "ro,barrier=1";
                fsmgr_flags = "wait";
            };
            fs_vendor: vendor {
                compatible = "android,vendor";
                dev = "/dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/vendor";
                type = "ext4";
                mnt_flags = "ro,barrier=1";
                fsmgr_flags = "wait";
            };
        };
    };
};

```

- 方案二：新增一个包含对应 dts 的文件，然后再覆盖 fsmgr\_flags 属性。如新增一个 dts 文件：sp9832e-1h10-gofu-avb.dts，详细修改参考下图。

```

1/*
2 * Spreadtrum SP9832E 1H10 GO FULL board DTS file
3 *
4 * Copyright (C) 2016-2017, Spreadtrum Communications Inc.
5 *
6 * This file is licensed under a dual GPLv2 or X11 license.
7 */
8#include "sp9832e-1h10-gofu.dts"
9
10&fs_system {
11     fsmgr_flags = "wait,avb";
12};
13
14&fs_vendor {
15     fsmgr_flags = "wait,avb";
16};

```

## 确认是否开启 Secure Boot

Dm-Verity 功能是 Secure Boot 功能的部分实现，它们用同一个宏开关控制。当 BOARD\_SECBOOT\_CONFIG := true 时，功能才会开启生效。

### 说明

- 在 Android 8.1 的 go 版本上 Dm-Verity 和 Secure Boot 默认都是关闭的（出于对性能影响的考虑，google 建议关闭此功能）。如需开启此功能，需要重新适配。
- 非 go 版本默认是开启的。

## 2.3.2 定制 key

Android 8.1 上 system 和 vendor 采用两个 key 分别签名。都在目录 vendor/sprd/proprieties-source/packimage\_scripts/signimage/sprd/config/下：

- rsa4096\_system.pem & rsa4096\_system\_pub.bin

- rsa4096\_vendor.pem & rsa4096\_vendor\_pub.bin

生成新 key 的方式：

步骤 1 进入目录：cd vendor/sprd/proprieties-source/packimage\_scripts/signimage/sprd/config

步骤 2 生成 system 分区的 key：Run ./genkey.sh system

步骤 3 生成 vendor 分区的 key：Run ./genkey.sh vendor

----结束

### 2.3.3 功能验证

验证 Dm-Verity 功能需要在 user 版本上验证，因为 userdebug 版本上该功能是关闭的。

验证功能是否成功开启的办法：查看 system 和 vendor 分区的挂载方式。当挂载方式为 dm-x 时，表明功能开启成功，如下图所示。反之则是未开启。

```
sp9832e_1h10:/ $ mount
rootfs on / type rootfs (ro,seclabel)
tmpfs on /dev type tmpfs (rw,seclabel,nosuid,relatime,mode=755)
devpts on /dev/pts type devpts (rw,seclabel,relatime,mode=600)
proc on /proc type proc (rw,relatime,gid=3009,hidepid=2)
sysfs on /sys type sysfs (rw,seclabel,relatime)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,relatime)
/dev/block/dm-0 on /system type ext4 (ro,seclabel,relatime,data=ordered)
/dev/block/dm-1 on /vendor type ext4 (ro,seclabel,relatime,data=ordered)
none on /acct type cgroup (rw,relatime,cpuacct)
debugfs on /sys/kernel/debug type debugfs (rw,seclabel,relatime)
tmpfs on /mnt type tmpfs (rw,seclabel,relatime,mode=755,gid=1000)
none on /config type configfs (rw,relatime)
```

### 2.3.4 功能关闭

修改 dts 文件中 system 和 vendor 分区对应的 fsmgr\_flags 值，去掉 “,avb”。

## 2.4 Android 9.0 平台适配

Dm-Verity 在 Android 9.0 采用 AVB2.0 的方案。

Android 9.0 上要求必须支持 System\_as\_root 功能。此功能要求 ramdisk.img 打包进 system.img。一旦 system.img 开启 Dm-Verity，system 分区的挂载将在 kernel 中进行，且挂载成 dm 设备的参数需要 uboot 解析，然后以 cmdline 的方式传递给 kernel。

Android 9.0 上增加了 product 分区，此分区也需要用 Dm-Verity 功能保护。

综上，system 分区将会通过解析 cmdline 的方式挂载。vendor、product 分区的挂载方式和 Android 8.1 保持一致，均由 dts 文件控制。

## 2.4.1 功能适配

### 配置 kernel config

Kuconfig 工具配置 Verity target support、Verity forward error correction support 两个功能：

```

--- Multiple devices driver support (RAID and LVM)
< > RAID support
< > Block device as cache
< * > Device mapper support
[ ] request-based DM: use blk-mq I/O path by default
[ ] Device mapper debugging support
[ ] Block manager locking
< * > Crypt target support
< > Snapshot target
< > Thin provisioning target
< > Cache target (EXPERIMENTAL)
< > Era target (EXPERIMENTAL)
< > Mirror target
< > RAID 1/4/5/6/10 target
< > Zero target
< > Multipath target
< > I/O delaying target
[ * ] DM uevents
< > Flakey target
< * > Verity target support
[ * ] Verity forward error correction support
< > Switch target support (EXPERIMENTAL)
< > Log writes target support
< > Integrity target support
< > Support AVB specific verity error behavior
[ ] Verity will validate blocks at most once
< > Backup block device

```

对应 deconfig 文件修改如下：

kernel/common / arch/arm/configs/sp9832a_2h11_volte_defconfig	Patch Set 1 2 3 4 (gitweb)	Patch Set 1 2 3 4 (gitweb)
1281 # CONFIG_BLK_DEV_MD is not set	1281 # CONFIG_BLK_DEV_MD is not set	1281 # CONFIG_BLK_DEV_MD is not set
1282 # CONFIG_BCACHE is not set	1282 # CONFIG_BCACHE is not set	1282 # CONFIG_BCACHE is not set
1283 CONFIG_BLK_DEV_DM_BUILTIN=y	1283 CONFIG_BLK_DEV_DM_BUILTIN=y	1283 CONFIG_BLK_DEV_DM_BUILTIN=y
1284 CONFIG_BLK_DEV_DM=y	1284 CONFIG_BLK_DEV_DM=y	1284 CONFIG_BLK_DEV_DM=y
1285 # CONFIG_DM_DEBUG is not set	1285 # CONFIG_DM_DEBUG is not set	1285 # CONFIG_DM_DEBUG is not set
1286 CONFIG_DM_CRYPT=y	1286 CONFIG_DM_CRYPT=y	1286 CONFIG_DM_CRYPT=y
1287 # CONFIG_DM_SNAPSHOT is not set	1287 # CONFIG_DM_SNAPSHOT is not set	1287 # CONFIG_DM_SNAPSHOT is not set
1288 # CONFIG_DM_THIN_PROVISIONING is not set	1288 # CONFIG_DM_THIN_PROVISIONING is not set	1288 # CONFIG_DM_THIN_PROVISIONING is not set
1289 # CONFIG_DM_CACHE is not set	1289 # CONFIG_DM_CACHE is not set	1289 # CONFIG_DM_CACHE is not set
1290 # CONFIG_DM_MIRROR is not set	1290 # CONFIG_DM_MIRROR is not set	1290 # CONFIG_DM_MIRROR is not set
1291 # CONFIG_DM_RAID is not set	1291 # CONFIG_DM_RAID is not set	1291 # CONFIG_DM_RAID is not set
1292 # CONFIG_DM_ZERO is not set	1292 # CONFIG_DM_ZERO is not set	1292 # CONFIG_DM_ZERO is not set
1293 # CONFIG_DM_MULTIPATH is not set	1293 # CONFIG_DM_MULTIPATH is not set	1293 # CONFIG_DM_MULTIPATH is not set
1294 # CONFIG_DM_DELAY is not set	1294 # CONFIG_DM_DELAY is not set	1294 # CONFIG_DM_DELAY is not set
1295 # CONFIG_DM_UEVENT is not set	1295 # CONFIG_DM_UEVENT is not set	1295 # CONFIG_DM_UEVENT is not set
1296 # CONFIG_DM_FLAKEY is not set	1296 # CONFIG_DM_FLAKEY is not set	1296 # CONFIG_DM_FLAKEY is not set
1297 # CONFIG_DM_VERITY is not set	1297 # CONFIG_DM_VERITY is not set	1297 # CONFIG_DM_VERITY is not set
1298 # CONFIG_TARGET_CORE is not set	1298 # CONFIG_TARGET_CORE is not set	1298 # CONFIG_TARGET_CORE is not set
1299 CONFIG_NETDEVICES=y	1299 CONFIG_NETDEVICES=y	1299 CONFIG_NETDEVICES=y
1300 CONFIG_NET_CORE=y	1300 CONFIG_NET_CORE=y	1300 CONFIG_NET_CORE=y
1301 # CONFIG_BONDING is not set	1301 # CONFIG_BONDING is not set	1301 # CONFIG_BONDING is not set

### 修改 dts 文件，开启 Dm-Verity flag

修改 dts 文件，适配 vendor 和 product 分区的挂载，并添加开启 Dm-Verity 的 flag。修改方案有两种：

- 方案一：system 和 vendor 的 fsmgr\_flags 从 “wait” 改成 “wait,avb”：

```

9/ {
10  firmware {
11      android {
12          compatible = "android,firmware";
13
14          vbmeta {
15              compatible = "android,vbmeta";
16              parts = "vbmeta,boot,recovery,system,vendor,product";
17          };
18          fstab {
19              compatible = "android,fstab";
20              vendor {
21                  compatible = "android,vendor";
22                  dev = "/dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/vendor";
23                  type = "ext4";
24                  mnt_flags = "ro,barrier=1";
25                  fsmgr_flags = "wait,avb";
26              };
27              product {
28                  compatible = "android,product";
29                  dev = "/dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/product";
30                  type = "ext4";
31                  mnt_flags = "ro,barrier=1";
32                  fsmgr_flags = "wait,avb";
33              };
34          };
35      };
36  };
37 };

```

- 方案二：新增一个包含对应 dts 的文件，然后再覆盖 fsmgr\_flags 属性。

– 原 dtsti 文件增加 fs\_vendor、fs\_product 标签：

<pre> 39  fstab { 40      &gt;&gt; compatible = "android,fstab"; 41      &gt;&gt; vendor { 42          &gt;&gt; compatible = "android,vendor"; 43          &gt;&gt; dev = "/dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/vendor"; 44          &gt;&gt; type = "ext4"; 45          &gt;&gt; mnt_flags = "ro,barrier=1"; 46          &gt;&gt; fsmgr_flags = "wait"; 47          &gt;&gt; }; 48      &gt;&gt; product { 49          &gt;&gt; compatible = "android,product"; 50          &gt;&gt; dev = "/dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/product"; 51          &gt;&gt; type = "ext4"; 52          &gt;&gt; mnt_flags = "ro,barrier=1"; 53          &gt;&gt; fsmgr_flags = "wait"; 54          &gt;&gt; }; 55  }; </pre>	<pre> 39  fstab { 40      &gt;&gt; compatible = "android,fstab"; 41      &gt;&gt; fs_vendor: vendor { 42          &gt;&gt; compatible = "android,vendor"; 43          &gt;&gt; dev = "/dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/vendor"; 44          &gt;&gt; type = "ext4"; 45          &gt;&gt; mnt_flags = "ro,barrier=1"; 46          &gt;&gt; fsmgr_flags = "wait"; 47          &gt;&gt; }; 48      &gt;&gt; fs_product: product { 49          &gt;&gt; compatible = "android,product"; 50          &gt;&gt; dev = "/dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/product"; 51          &gt;&gt; type = "ext4"; 52          &gt;&gt; mnt_flags = "ro,barrier=1"; 53          &gt;&gt; fsmgr_flags = "wait"; 54          &gt;&gt; }; 55  }; </pre>
---	--

- 新增一个 dts 文件，引用原 dtsi 文件定义的标签，覆盖 fsmgr\_flags 属性：

```
1 /*
2  * Spreadtrum SP9832E 1H10 GO FULL board DTS file
3  *
4  * Copyright (C) 2016-2017, Spreadtrum Communications Inc.
5  *
6  * This file is licensed under a dual GPLv2 or X11 license.
7  */
8 #include "sp9832e-1h10-gofu.dts"
9
10 &fs_system {
11     fsmgr_flags = "wait, avb";
12 };
13
14 &fs_vendor {
15     fsmgr_flags = "wait, avb";
16 };
```

## 确认是否开启 Secure Boot

Dm-Verity 功能是 Secure Boot 功能的部分实现，它们用同一个宏开关控制。即 BOARD\_SECBOOT\_CONFIG := true 时，功能才会开启生效。

### 说明

- 在 Android 9.0 的 go 版本上此功能默认都是关闭的（出于对性能影响的考虑，google 建议关闭此功能）。如需要开启此功能，需要重新适配。
- 非 go 版本默认是开启的。

## 2.4.2 定制 key

Android 9.0 上 key 的配置和 Android 8.1 原理上一致。

Android 9.0 上增加了 product 分区，即需要增加 product 分区的 key。system 和 vendor 两分区的 key 和 Android 8.1 保持一致，请参考“2.2.2 定制 key”。

## 2.4.3 功能验证

验证功能是否成功开启的办法：查看 system、vendor、product 分区的挂载方式。

通过 mount 命令查看分区挂载。如果 vendor、product 分别挂载成 dm-2、dm-1，说明 vendor、product 已经挂载成功，如下图所示。

```
s9863a1h10:/ # mount
/dev/root on / type ext4 (ro,seclabel,relatime,block_validity,delalloc,barrier,user_xattr)
devtmpfs on /dev type devtmpfs (rw,seclabel,relatime,size=928860k,nr_inodes=232215,mode=755)
tmpfs on /dev type tmpfs (rw,seclabel,nosuid,relatime,mode=755)
devpts on /dev/pts type devpts (rw,seclabel,relatime,mode=600)
proc on /proc type proc (rw,relatime,gid=3009,hidepid=2)
sysfs on /sys type sysfs (rw,seclabel,relatime)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,relatime)
tmpfs on /mnt type tmpfs (rw,seclabel,nosuid,nodev,noexec,relatime,mode=755,gid=1000)
/dev/block/dm-1 on /product type ext4 (ro,seclabel,relatime,block_validity,delalloc,barrier,user_xattr)
/dev/block/dm-2 on /vendor type ext4 (ro,seclabel,relatime,block_validity,delalloc,barrier,user_xattr)
none on /acct type cgroup (rw,nosuid,nodev,noexec,relatime,cpuacct)
debugfs on /sys/kernel/debug type debugfs (rw,seclabel,relatime)
none on /dev/stune type cgroup (rw,nosuid,nodev,noexec,relatime,schedtune)
none on /config type configs (rw,nosuid,nodev,noexec,relatime)
none on /dev/cpuctl type cgroup (rw,nosuid,nodev,noexec,relatime,cpu)
none on /dev/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset,noprefix,release_agent=/sbin/cpuset_release_agent)
pstore on /sys/fs/pstore type pstore (rw,seclabel,nosuid,nodev,noexec,relatime)
adb on /dev/usb-lun0 type functionfs (rw,relatime)
tracefs on /sys/kernel/debug/tracing type tracefs (rw,seclabel,relatime)
/dev/block/mmcblk0p31 on /cache type ext4 (rw,seclabel,nosuid,nodev,noatime,noauto_da_alloc,data=ordered)
/dev/block/mmcblk0p1 on /mnt/vendor type ext4 (rw,seclabel,nosuid,nodev,noatime,noauto_da_alloc,data=ordered)
tmpfs on /storage type tmpfs (rw,seclabel,nosuid,nodev,noexec,relatime,mode=755,gid=1000)
cgroup_root on /sys/fs/cgroup type tmpfs (rw,seclabel,relatime)
cgroup_root on /sys/fs/cgroup type tmpfs (rw,seclabel,relatime)
```

system 分区的挂载无法通过 mount 命令查看。如果查看到/dev/block/dm-0，说明 system 已经挂载成 dm 设备，如下图所示。

```
s9863a1h10:/dev/block # ls -l
total 0
drwxr-xr-x 2 root root 760 2018-11-14 19:06 by-name
brw----- 1 root root 253, 0 2018-11-14 19:06 dm-0
brw----- 1 root root 253, 1 2018-11-14 19:06 dm-1
brw----- 1 root root 253, 2 2018-11-14 19:06 dm-2
brw----- 1 root root 253, 3 2018-11-14 19:07 dm-3
```

综上，3 个分区都挂载成功时，Dm-Verity 功能开启成功。

## 2.4.4 功能关闭

从 Android 9.0 开始，在 user、userdebug 版本上都开启了 Dm-Verity 功能。

在 Secure Boot 功能开启时（即 BOARD\_SECBOOT\_CONFIG := true），单独关闭 Dm-Verity 功能的方法如下：

- vendor 和 product 分区关闭 Dm-Verity 的方法：修改 dts 文件中分区对应的 fsmgr\_flags 值，去掉“avb”。
- system 分区关闭 Dm-Verity 的方法：PRODUCT\_DMVERITY\_DISABLE := true。

## 2.5 Android 10.0 平台适配

Android 10.0 较之前版本存在以下两点变化：

- 新增了动态分区功能，将 system、vendor 和 product 分区打包进 super 分区，挂载的时候以逻辑分区的形式挂载。因此开机启动 bootloader 阶段无法读取用户空间动态分区的 vbmeta footer 信息。在 Android 10.0 以前的版本上，该信息存储在分区镜像文件中。在 Android 10.0 上，这部分信息被移到对应的 vbmeta 分区中。所以 Android 10.0 新增了两个 vbmeta 分区：
  - vbmeta\_system 存储 system、product 分区的 vbmeta 信息。
  - vbmeta\_vendor 存储 vendor 分区的 vbmeta 信息。
- BSP 独立编译新增了 socko、odmko 两个分区，这两个分区也需要进行 Dm-Verity 校验。

## 2.5.1 功能适配

### 配置 kernel config

Kuconfig 工具配置 Verity target support、Verity forward error correction support 两个功能：

```
--- Multiple devices driver support (RAID and LVM)
< > RAID support
< > Block device as cache
<*> Device mapper support
[ ] request-based DM: use blk-mq I/O path by default
[ ] Device mapper debugging support
[ ] Block manager locking
<*> Crypt target support
< > Snapshot target
< > Thin provisioning target
< > Cache target (EXPERIMENTAL)
< > Era target (EXPERIMENTAL)
< > Mirror target
< > RAID 1/4/5/6/10 target
< > Zero target
< > Multipath target
< > I/O delaying target
[*] DM uevents
< > Flakey target
<*> Verity target support
[*] Verity forward error correction support
< > Switch target support (EXPERIMENTAL)
< > Log writes target support
< > Integrity target support
< > Support AVB specific verity error behavior
[ ] Verity will validate blocks at most once
< > Backup block device
```

对应 deconfig 文件修改如下：

kernel/common / arch/arm/configs/sp9832a_2h11_volte_defconfig	Patch Set 1 2 3 4 (gitweb)
1281 # CONFIG_BLK_DEV_MD is not set	1281 # CONFIG_BLK_DEV_MD is not set
1282 # CONFIG_BCACHE is not set	1282 # CONFIG_BCACHE is not set
1283 CONFIG_BLK_DEV_DM_BUILTIN=y	1283 CONFIG_BLK_DEV_DM_BUILTIN=y
1284 CONFIG_BLK_DEV_DM=y	1284 CONFIG_BLK_DEV_DM=y
1285 # CONFIG_DM_DEBUG is not set	1285 # CONFIG_DM_DEBUG is not set
1286 CONFIG_DM_CRYPT=y	1286 CONFIG_DM_CRYPT=y
1287 # CONFIG_DM_SNAPSHOT is not set	1287 # CONFIG_DM_SNAPSHOT is not set
1288 # CONFIG_DM_THIN_PROVISIONING is not set	1288 # CONFIG_DM_THIN_PROVISIONING is not set
1289 # CONFIG_DM_CACHE is not set	1289 # CONFIG_DM_CACHE is not set
1290 # CONFIG_DM_MIRROR is not set	1290 # CONFIG_DM_MIRROR is not set
1291 # CONFIG_DM_RAID is not set	1291 # CONFIG_DM_RAID is not set
1292 # CONFIG_DM_ZERO is not set	1292 # CONFIG_DM_ZERO is not set
1293 # CONFIG_DM_MULTIPATH is not set	1293 # CONFIG_DM_MULTIPATH is not set
1294 # CONFIG_DM_DELAY is not set	1294 # CONFIG_DM_DELAY is not set
1295 # CONFIG_DM_UEVENT is not set	1295 # CONFIG_DM_UEVENT is not set
1296 # CONFIG_DM_FLAKEY is not set	1296 # CONFIG_DM_FLAKEY is not set
1297 # CONFIG_DM_VERITY is not set	1297 # CONFIG_DM_VERITY=y
1298 # CONFIG_TARGET_CORE is not set	1298 # CONFIG_TARGET_CORE is not set
1299 CONFIG_NETDEVICES=y	1299 CONFIG_NETDEVICES=y
1300 CONFIG_NET_CORE=y	1300 CONFIG_NET_CORE=y
1301 # CONFIG_BONDING is not set	1301 # CONFIG_BONDING is not set

### fstab 增加 avb flag

system/vendor/product 分区开启 Dm-Verity 方法如下：



为了适配动态分区和 SYSTEM AS ROOT 功能，Android 10.0 新增了 fstab.ramdisk。该 fstab 主要负责内存文件系统挂载、动态分区解析和挂载，system、vendor 和 product 分区 Dm-Verity 校验标志也在该文件中配置。

由于新增了 vbmeta\_system、vbmeta\_vendor 两个分区，“avb=” flag 需要指明 vbmeta 信息存储在哪个分区。UNISOC 方案中 vbmeta\_system 存储 system、product 分区的 vbmeta 信息，vbmeta\_vendor 存储 vendor 分区的 vbmeta 信息。具体修改如下：

<pre>1 #Dynamic partitions fstab file 2 #&lt;dev&gt; &lt;mnt_point&gt; &lt;type&gt; &lt;mnt_flags options&gt; &lt;fs_mgr_flags&gt; 3 4 system /system ext4 ro,barrier=1 wait,logical,first_stage_mount 5 vendor /vendor ext4 ro,barrier=1 wait,logical,first_stage_mount 6 product /product ext4 ro,barrier=1 wait,logical,first_stage_mount 7 /dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/metadata /metadata ext4 nodev, noatime, nosuid</pre>	<pre>1 #Dynamic partitions fstab file 2 #&lt;dev&gt; &lt;mnt_point&gt; &lt;type&gt; &lt;mnt_flags options&gt; &lt;fs_mgr_flags&gt; 3 4 system /system ext4 ro,barrier=1 wait,avb=vbmeta_system,logical,first_stage_mount 5 vendor /vendor ext4 ro,barrier=1 wait,avb=vbmeta_vendor,logical,first_stage_mount 6 product /product ext4 ro,barrier=1 wait,avb=vbmeta,logical,first_stage_mount 7 /dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/metadata /metadata ex</pre>
--	--

socko/odmko 分区开启 Dm-Verity 方法如下：

由于 socko、odmko 并非动态分区，其 vbmeta 信息仍存储在对应分区中，“avb=” flag 指向对应分区即可。具体修改如下：

<pre>22 /mnt/vendor/socko ext4 ro, noatime, nosuid, nodev, noulk_io_submit, noauto_da_alloc wait, check 23 /mnt/vendor/odmko ext4 ro, noatime, nosuid, nodev, noulk_io_submit, noauto_da_alloc wait, check</pre>	<pre>22 /mnt/vendor/socko ext4 ro, noatime, nosuid, nodev, noulk_io_submit, noauto_da_alloc wait, avb=socko, c 23 /mnt/vendor/odmko ext4 ro, noatime, nosuid, nodev, noulk_io_submit, noauto_da_alloc wait, avb=odmko, c</pre>
--	--

## fstab 增加 avb\_keys flag

Android 10.0 GSI 版本上，Google 原生 system 镜像也会做 Dm-Verity 校验。但需要增加 “avb\_keys=” flag，指明 GSI system 镜像 key 的存储路径，否则 GSI 版本会由于挂载 system 分区失败导致无法开机。需要在 fstab.ramdisk 中，system 分区那一行增加以下 flag：

```
avb_keys=/avb/q-gsi.avbpubkey:/avb/r-gsi.avbpubkey:/avb/s-gsi.avbpubkey
```

具体修改如下：

```
#Dynamic partitions fstab file
#<dev> <mnt_point> <type> <mnt_flags options> <fs_mgr_flags>

system /system ext4 ro,barrier=1 wait,avb=vbmeta_system,logical,first_stage_mount,avb_keys=/avb/q-gsi.avbpubkey:/avb/r-gsi.avbpubkey:/avb/s-gsi.avbpubkey
vendor /vendor ext4 ro,barrier=1 wait,avb=vbmeta_vendor,logical,first_stage_mount
product /product ext4 ro,barrier=1 wait,avb=vbmeta,logical,first_stage_mount
/dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/metadata /metadata ext4 nodev, noatime, nosuid, errors=panic wait, formattable, first_stage_mount
```

## 2.5.2 定制 key

Android 10.0 上 key 的配置方法和 Android 8.1、Android 9.0 保持一致。Android 10.0 共有 system、vendor、product、socko 和 odmko 五个分区开启了 Dm-Verity 校验，每个分区都需要生成对应的 key。具体方法如下：

- 步骤 1 进入目录：cd vendor/sprd/proprietary-source/packimage\_scripts/signimage/sprd/config
- 步骤 2 生成 system 分区的 key：Run ./genkey.sh system
- 步骤 3 生成 product 分区的 key：Run ./genkey.sh product
- 步骤 4 生成 vendor 分区的 key：Run ./genkey.sh vendor
- 步骤 5 生成 socko 分区的 key：Run ./genkey.sh socko
- 步骤 6 生成 odmko 分区的 key：Run ./genkey.sh odmko

----结束

执行上述 genkey.sh 脚本后，每个分区都会生成 rsa4096\_XXX.pem、rsa4096\_XXX\_pub.bin 两个文件。



## 2.5.3 功能验证

### 确认 system、vendor、product 分区是否开启 Dm-Verity 校验

Android 10.0 新增了基于 Linux 内核 dm-linear 的动态分区功能。不开启 Dm-Verity 时，system、vendor 和 product 分区也会挂载为 dm 设备。因此这三个分区无法通过查看设备挂载状态确认。可以通过查看串口 log，分区挂载前是否构建 Dm-Verity 映射表，确认是否开启 Dm-Verity。具体 log 信息如下：

- system 分区

```
init: [libfs_avb]Built verity table: '1 /dev/block/dm-0 /dev/block/dm-0 4096 4096 313614 313614
sha1 4838d910705aeabeba55416e26863500bc7eec0b e31af2081be98ea257df30db080d570b0cad5b5 10
use_fec_from_device /dev/block/dm-0 fec_roots 2 fec_blocks 316086 fec_start 316086
restart_on_corruption ignore_zero_blocks'
device-mapper: verity: sha1 using implementation "sha1-ce"
init: [libfs_mgr]__mount(source=/dev/block/dm-3,target=/system,type=ext4)=0: Success
device-mapper: verity: sha1 using implementation "sha1-ce"
```

- vendor 分区

```
init: [libfs_avb]Built verity table: '1 /dev/block/dm-1 /dev/block/dm-1 4096 4096 82470 82470
sha1 230d7d714a21f62c84f7306ff7a467533c761243 d9a9ce03416b28207ac29e50a9bbd3546bbfe154 10
use fec from device /dev/block/dm-1 fec_roots 2 fec_blocks 83122 fec_start 83122
restart_on_corruption ignore_zero_blocks'
device-mapper: verity: sha1 using implementation "sha1-ce"
init: [libfs_mgr]__mount(source=/dev/block/dm-4,target=/vendor,type=ext4)=0: Success
```

- product 分区

```
init: [libfs_avb]Built verity table: '1 /dev/block/dm-2 /dev/block/dm-2 4096 4096 360010 360010
sha1 5f28681e9ae5c356dbbcc7f4c632a8301fcf612c 0fe99a5dfc2297cd2294f7c880124b87c4bc042a 10
use fec from device /dev/block/dm-2 fec_roots 2 fec_blocks 362846 fec_start 362846
restart_on_corruption ignore_zero_blocks'
device-mapper: verity: sha1 using implementation "sha1-ce"
init: [libfs_mgr]__mount(source=/dev/block/dm-5,target=/product,type=ext4)=0: Success
```

### 确认 socko、odmko 分区是否开启 Dm-Verity 校验

新增的 socko 和 odmko 分区由于不是动态分区，仍可通过是否挂载为 dm 设备判断是否开启 Dm-Verity。执行 mount 命令，socko 和 odmko 分区挂载状态如下：

```
/dev/block/dm-6 on /mnt/vendor/socko type ext4 (ro,seclabel,nosuid,nodev,noatime
,noauto_da_alloc)
/dev/block/dm-7 on /mnt/vendor/odmko type ext4 (ro,seclabel,nosuid,nodev,noatime
,noauto_da_alloc)
```

### 确认使用的哈希算法

哈希运算使用 SHA 算法，对于不支持 ARM-CE 的芯片，使用 sha1-neon 实现，串口 log 打印如下：

```
device-mapper: verity: sha1 using implementation "sha1-neon"
```

对于支持 ARM-CE 的芯片，使用 sha1-ce 实现，串口 log 打印如下：

```
device-mapper: verity: sha1 using implementation "sha1-ce"
```

UNISOC 目前支持 Dm-Verity 的量产芯片中，除 SC7731E 不支持 ARM-CE 外，其余芯片均支持。需根据如上串口 log 来确认使用的算法。对于支持 ARM-CE 的芯片，sha1-ce 性能优于 sha1-neon。

## 2.5.4 功能关闭

### 量产版本关闭 Dm-Verity 校验

Google CDD 文档要求开启 Verified Boot 功能，所以对于需要过 Google 认证的项目，都需要开启 Dm-Verity 校验。但如果有关闭该功能的需求，也支持通过以下方式关闭：

步骤 1 找到对应 board 的 fstab 文件：fstab.ramdisk、fstab.xxx、fstab.xxx.f2fs.....

步骤 2 删除所有“avb=vbmeta\_system”、“avb=vbmeta\_vendor”、“avb=vbmeta” flag。

步骤 3 删除 system 分区的“avb\_keys=/avb/q-gsi.avbpubkey:/avb/r-gsi.avbpubkey:/avb/s-gsi.avbpubkey”。

步骤 4 重新编译版本。

----结束

### 临时关闭 Dm-Verity 校验

如调试需要临时关闭或需要执行 remount 操作，可先根据《Android10.0 设备解锁指导手册》文档解锁设备后，执行以下命令：

```
$ adb root
$ adb disable-verity
$ adb reboot
$ adb wait-for-device
$ adb root
$ adb remount
```

## 2.6 Android 11.0 平台适配

Android 11.0 新增了一个动态分区 system\_ext，它和 system、vendor、product 分区一起打包进 super 分区，挂载的时候以逻辑分区的形式挂载。与 Android 10.0 相同，由于开机启动 bootloader 阶段无法读取用户空间动态分区，需要将这几个动态分区的 vbmeta 信息存储在单独的 vbmeta 分区。Android11.0 上，每个动态分区分别对应一个 vbmeta 分区，具体如下表：

动态分区	vbmeta 分区
system	system_vbmeta
system_ext	system_ext_vbmeta
vendor	vendor_vbmeta
prodect	product_vbmeta

通过下载工具重新下载 super 分区，或者通过 fastbootd 命令重新下载动态分区后，需要同步更新相应的 vbmeta 分区。

## 2.6.1 功能适配

### 配置 kernel config

kuconfig 工具配置 Verity target support、Verity forward error correction support、Support AVB specific verity error behavior 三个功能：

```
--- Multiple devices driver support (RAID and LVM)
< > RAID support
< > Block device as cache
< * > Device mapper support
[ ] request-based DM: use blk-mq I/O path by default
[ ] Device mapper debugging support
[ ] Block manager locking
< * > Crypt target support
< * > Default-key target support
< * > Snapshot target
< > Thin provisioning target
< > Cache target (EXPERIMENTAL)
< > Era target (EXPERIMENTAL)
< > Mirror target
< > RAID 1/4/5/6/10 target
< > Zero target
< > Multipath target
< > I/O delaying target
[ * ] DM uevents
< > Flakey target
< * > Verity target support
[ * ] Verity forward error correction support
< > Switch target support (EXPERIMENTAL)
< > Log writes target support
< > Integrity target support
< * > Support AVB specific verity error behavior
[ ] Android verity target support
[ ] Verity will validate blocks at most once
< * > Backup block device
```

### fstab 增加 avb flag

system、system\_ext、vendor、product 分区开启 Dm-Verity 方法如下：

与 Android10.0 相同，为了适配动态分区和 SYSTEM AS ROOT 功能，fstab.ramdisk 主要负责内存文件系统挂载、动态分区解析和挂载，system、system\_ext、vendor 和 product 分区的 Dm-Verity 校验标志也在该文件中配置。同时“avb=”标签需要指明动态分区的 vbmeta 信息存储在哪个 vbmeta 分区。以 UMS512 为例，可以参考 IDH 包中的以下文件配置：

/sprdroidr\_trunk/device/sprd/mpool/module/partition/msoc/sharkl5Pro/fstab.ramdisk

socko、odmko 分区开启 Dm-Verity 方法如下：

由于 socko、odmko 并非动态分区，其 vbmeta 信息仍存储在对应分区中，“avb=”标签指向对应分区即可。以 UMS512 为例，可以参考 IDH 包中的以下文件配置：

/sprdroidr\_trunk/device/sprd/mpool/module/partition/msoc/sharkl5Pro/fstab

## fstab 增加 avb\_keys flag

Android 11.0 GSI 版本上, Google 原生 system 镜像也会做 Dm-Verity 校验。但需要增加 “avb\_keys=” 标签, 指明 GSI system 镜像 key 的存储路径, 否则 GSI 版本会由于挂载 system 分区失败导致无法开机。具体修改需要在 fstab.ramdisk 中, system 分区那一行增加以下 flag:

```
avb_keys=/avb/q-gsi.avbpubkey:/avb/r-gsi.avbpubkey:/avb/s-gsi.avbpubkey
```

## 2.6.2 定制 key

Android 11.0 上 key 的配置方法和 Android 8.1、Android 9.0 以及 Android 10.0 保持一致。Android 11.0 共有 system、system\_ext、vendor、product、socko 和 odmko 六个分区开启了 Dm-Verity 校验, 每个分区都需要生成对应的 key, 具体方法如下:

步骤 1 进入目录: `cd vendor/sprd/proprieties-source/packimage_scripts/signimage/sprd/config`

步骤 2 生成 system 分区的 key: Run `./genkey.sh system`

步骤 3 生成 system\_ext 分区的 key: Run `./genkey.sh system_ext`

步骤 4 生成 product 分区的 key: Run `./genkey.sh product`

步骤 5 生成 vendor 分区的 key: Run `./genkey.sh vendor`

步骤 6 生成 socko 分区的 key: Run `./genkey.sh socko`

步骤 7 生成 odmko 分区的 key: Run `./genkey.sh odmko`

步骤 8 执行上述 genkey.sh 脚本后, 每个分区都会生成 rsa4096\_xxx.pem、rsa4096\_xxx\_pub.bin 两个文件, 将这些文件预置在 vendor/sprd/proprieties-source/packimage\_scripts/signimage/sprd/config/目录。

----结束

## 2.6.3 功能验证

### 确认 system、system\_ext、vendor、product 分区是否开启 Dm-Verity 校验

由于启用了动态分区功能, 该功能基于 Linux 内核 dm-linear。不开启 Dm-Verity 时, system、system\_ext、vendor 和 product 分区也会挂载为 dm 设备, 因此这四个分区无法通过查看设备挂载状态确认。可以通过 dmctl 命令获取 Dm-Verity 映射表确认:

- system 分区

```
$ adb root
$ adb shell dmctl table system-verity
```

输出:

```
Targets in the device-mapper table for system-verity:
0-1100832: verity, 1 252:0 252:0 4096 4096 137604 137604 sha1
586ff4c1db6b0b435718090bf03c3e7de1dc707a 79ee3646176a38b29018a0aaf7dd1d92d0502d1d 11
restart on corruption ignore zero blocks check at most once use fec from device 252:0 fec blocks
138690 fec_start 138690 fec_roots 2
```

- system\_ext 分区

```
$ adb root
$ adb shell dmctl table system_ext-verity
```

输出:

```
Targets in the device-mapper table for system_ext-verity:
0-550464: verity, 1 252:1 252:1 4096 4096 68808 68808 sha1
da34fff41964cfbdc53811fb00ad315708657385 d2dded84053f72116b559c7e358cc0278a41026a 11
restart on corruption ignore zero blocks check at most once use fec from device 252:1 fec blocks
69352 fec_start 69352 fec_roots 2
```

- vendor 分区

```
$ adb root
$ adb shell dmctl table vendor-verity
```

输出:

```
Targets in the device-mapper table for vendor-verity:
0-368312: verity, 1 252:2 252:2 4096 4096 46039 46039 sha1
c21d791ac53181d198588d992d9250828453cfb6 69d1c37b2bc63759af2b0b306ce0feba59e147e8 11
restart_on_corruption ignore_zero_blocks check_at_most_once use_fec_from_device 252:2 fec_blocks
46403 fec_start 46403 fec_roots 2
```

- product 分区

```
$ adb root
$ adb shell dmctl table product-verity
```

输出:

```
Targets in the device-mapper table for product-verity:
0-1415920: verity, 1 252:3 252:3 4096 4096 176990 176990 sha1
f4b29d17766b015e94ce03bb2baa75ee769db5e7 35dc3648baad4450b60b45a4bbff96a9c9c354da 11
restart on corruption ignore zero blocks check at most once use fec from device 252:3 fec blocks
178385 fec_start 178385 fec_roots 2
```

## 确认 socko、odmko 分区是否开启 Dm-Verity 校验

新增的 **socko** 和 **odmko** 分区由于不是动态分区，仍可通过是否挂载为 **dm** 设备判断是否开启 **Dm-Verity**。执行 **mount** 命令，**socko** 和 **odmko** 分区挂载状态如下：

```
/dev/block/dm-8 on /mnt/vendor/socko type ext4 (ro,seclabel,nosuid,nodev,noatime,noauto_da_alloc)
/dev/block/dm-9 on /mnt/vendor/odmko type ext4 (ro,seclabel,nosuid,nodev,noatime,noauto_da_alloc)
```

## 确认使用的哈希算法

哈希运算使用 **SHA** 算法，对于不支持 **ARM-CE** 的芯片，使用 **sha1-neon** 实现，串口 **log** 打印如下：

```
device-mapper: verity: sha1 using implementation "sha1-neon"
```

对于支持 **ARM-CE** 的芯片，使用 **sha1-ce** 实现，串口 **log** 打印如下：

```
device-mapper: verity: sha1 using implementation "sha1-ce"
```

UNISOC 目前支持 **Dm-Verity** 的量产芯片中，除 **SC7731E** 不支持 **ARM-CE** 外，其余芯片均支持。需根据如上串口 **log** 来确认使用的算法。对于支持 **ARM-CE** 的芯片，**sha1-ce** 性能优于 **sha1-neon**。

## 2.6.4 功能关闭

### 量产版本关闭 Dm-Verity 校验

Google CDD 文档要求开启 Verified Boot 功能，所以对于需要过 Google 认证的项目，都需要开启 Dm-Verity 校验。但如果有关闭该功能的需求，也支持通过以下方式关闭：

步骤 1 找到对应 board 的 fstab 文件：fstab.ramdisk、fstab.xxx、fstab.xxx.f2fs……

步骤 2 删除所有“avb=vbmata\_system”、“avb=vbmata\_vendor”、“avb=vbmata” flag。

步骤 3 删除 system 分区的“avb\_keys=/avb/q-gsi.avbpubkey:/avb/r-gsi.avbpubkey:/avb/s-gsi.avbpubkey”。

步骤 4 重新编译版本

----结束

### 临时关闭 Dm-Verity 校验

与 Android 10.0 不同，Android 11.0 如调试需要执行 remount 操作。在解锁 bootloader 后，执行以下命令即可，不再需要单独执行 adb disable-verity：

```
$ adb root
$ adb remount
```

# 3

## 常见问题

### 3.1 性能影响

开启 Dm-Verity 功能后，增加了校验逻辑分区 Block 数据流程，会对性能造成一定的影响。根据 Dm-Verity 的工作原理，对性能的影响主要体现在两个方面：

- 开机挂载 system、vendor 和 product 分区前，需要对相应的 vbmeta footer 数据进行校验。挂载分区后，系统启动过程从 system、vendor 和 product 分区读取数据都要进行 Hash 校验，增加耗时预计在 2~3s 左右。
- 系统运行时，从 system、vendor 和 product 等只读分区读取的数据都需要进行 hash 校验，因此对只读分区的 IO 会有一定影响。

### 3.2 OTA 影响

当使能 Dm-Verity 之后，需要将 OTA 升级方式切换为基于 Block 的升级方式。即 OTA 更新要在块设备层操作，并同时更新哈希树和 Verity metadata。

### 3.3 Android 9.0 userdebug 版本 remount 失败

Android 9.0 上在非 go 的项目上都是开启了 Secure Boot 和 Dm-Verity 功能的，在 go 的项目上只开启了 Secure Boot 功能，未开启 Dm-Verity 功能。

因此在非 go 项目的 userdebug 版本上直接使用以下命令会失败。

```
adb root
adb remount
```

因为一旦开启 Dm-Verity 功能去保护 system 分区，则 system 无法被 remount 成可读写的分区。

解决方案：首先需要 unlock bootloader，然后再 remount 手机即可成功。（unlock bootloader 的详细操作请参考文档《Android9.0 Unlock Bootloader Guide》）

为了方便 debug 的临时方案：在 userdebug 版本中对应的 board 中临时定义 `PRODUCT_DMVERITY_DISABLE := true` 可关闭 Dm-Verity 功能，这样可以直接 remount。但后续量产版本切记要将上述临时修改回退。

如需要在 userdebug 版本上单独关闭 Dm-Verity 功能，可通过用 userdebug 的判断条件控制 `PRODUCT_DMVERITY_DISABLE`。参考修改如下：

```
ifeq ($(TARGET_BUILD_VARIANT),userdebug)
    PRODUCT_DMVERITY_DISABLE := true
endif
```

## 3.4 功能异常排查指引

Dm-Verity 功能在平台参考 Board 上已做适配。若新增 Board 配置后出现无法开机等异常，可以参照以下步骤排查：

步骤 1 按照平台版本，检查本文档各项配置是否正确，尤其注意 Board 的适配。

步骤 2 检查编译 log，确认是否编译出 build\_verity\_tree 和 build\_verity\_metadata 信息。

步骤 3 串口抓取开机 kernel.log，检查 system&vendor 分区是否挂载正常，关键字：fs\_mgr。

步骤 4 针对 Android 7.0，需要确认 pac 包中 system.img 的大小和 xml 分区表上 system 分区的大小是否一致。

----结束



# 4

## 参考文档

---

《Android9.0 Unlock Bootloader Guide》

《Android10.0 设备解锁指导手册》