TNG TECHNOLOGY CONSULTING

DATENSCHUTZ IN ZEITEN VON BIG DATA

Moderne Methoden für gesetzeskonformen Datenschutz
im Kontext von Customer Analytics und Big Data

Autoren: Maximilian Bode, Dr. Daniel Kraschewski, Michael Pisula,
Christoph Stock, Dr. Mayra Stuhldreier und Prof. Dr. Gregor Thüsing

DATENSCHUTZ IN ZEITEN VON BIG DATA

VORWORT

Christoph Stock

Liebe Leserin, lieber Leser,

ab dem 25. Mai 2018 gilt die neue EU-Datenschutz-Grundverordnung in allen EU-Mitgliedsländern. Sie wurde unter anderem nach dem Vorbild des deutschen Datenschutzrechts entworfen, welches international als besonders restriktiv gilt. In einigen Punkten geht sie sogar noch darüber hinaus – damit wird deutlich, dass Datenschutz in Zukunft mindestens europaweit eine bedeutende Rolle spielen wird. Auf der anderen Seite wächst in Zeiten von Google, Facebook und Co. auch bei vielen deutschen und europäischen Unternehmen stetig das Interesse, die allgegenwärtigen persönlichen Daten kommerziell zu nutzen.

Analysen von Kundendaten ermöglichen gezieltes Marketing und bedarfsgesteuerte Produktentwicklung – womit sich etwa sinkende Margen in etablierten Geschäftsbereichen kompensieren oder auch völlig neuartige Geschäftsfelder erschließen lassen. Neben dem unmittelbaren wirtschaftlichen Nutzen kann umfassende Datenauswertung aber auch gesamtgesellschaftliche Vorteile mit sich bringen: So sind Bewegungsdaten aus Mobilfunknetzen für Straßen- und Verkehrsplanung von großem Interesse und medizinische Forschung wäre ohne Studien auf Basis von personenbezogenen Informationen gar nicht möglich.

Glücklicherweise schließen sich Datenschutz und Datennutzung nur auf den ersten Blick gegenseitig aus. Aktuelle Big-Data-Technologien in Verbindung mit modernen kryptographischen Verfahren, die weit über einfache Verschlüsselung hinausgehen, ermöglichen umfassende Analysen und gewinnbringende Nutzung großer Datenmengen ohne Verletzung der Privatsphäre des Einzelnen. Die Entwicklung konkreter Lösungen in diesem Umfeld ist keine leichte Aufgabe, aber die künftigen restriktiven Datenschutzregeln stellen auch eine Chance dar: „Privacy made in Europe" hat das Potential, zu einem internationalen Gütesiegel zu werden.

Anwendungen, die mit sensiblen personenbezogenen Daten arbeiten, sollten von vornherein mit einem starken Fokus auf Datenschutz und Sicherheit entwickelt werden. Der nachträgliche Einbau von Schutzmaßnahmen ist oft schwierig bis unmöglich und endet in der Regel entweder mit einer kompletten Neuentwicklung oder einem lückenhaften Ergebnis. Das Prinzip von „Privacy by Design" ist deshalb auch in der neuen EU-Datenschutz-Grundverordnung fest verankert. Ganz abgesehen von der Gesetzgebung sollten Firmen aber bereits aus eigenem Antrieb proaktiv den Schutz von Kundendaten anstreben. Denn so schwer es ist, eine vertrauensvolle Kundenbeziehung aufzubauen, so schnell verliert man dieses Vertrauen wieder. Gerade Software, die personenbezogene Daten analysiert, muss nach höchsten Standards erstellt werden. Während bei anderen Projekten Programmierfehler oder undurchdachte Architekturentscheidungen vor allem zu Verzögerungen und zusätzlichen Entwicklungskosten führen, steht in diesen Fällen deutlich mehr auf dem Spiel – nicht zuletzt auch durch die rechtlichen Konsequenzen.

Als Dienstleister für IT-Lösungen mit höchstem Qualitätsanspruch ist die TNG Technology Consulting GmbH deshalb in diesem hochsensiblen Bereich aktiv und unterstützt ihre Kunden bei der erfolgreichen Umsetzung entsprechender Projekte. Einige der Erfahrungen, die wir in diesen Projekten gesammelt haben, sowie Know-how zum aktuellen Stand bei Technologien und moderner Kryptographie möchten wir mit Ihnen in diesem Whitepaper teilen. Um auch die juristischen Aspekte von Datenschutz im Big-Data-Kontext angemessen zu beleuchten, konnten wir zudem mit Prof. Dr. Thüsing von der Universität Bonn einen der führenden Experten auf diesem Gebiet für ein Interview gewinnen.

Wir wünschen eine spannende und aufschlussreiche Lektüre!

Christoph Stock
Managing Partner, TNG Technology Consulting GmbH

Vorwort **3**

Interview mit Prof. Dr. Thüsing **5**

1 Methoden zur datenschutzkonformen Auswertung personenbezogener Informationen **8**

1.1 Grenzen klassischer IT-Security und darüber hinausgehende Mittel der modernen Kryptographie 9

1.2 Design-Prinzipien für datenschutzkritische Applikationen 14

1.3 Notwendigkeit, Stärken und Schwächen formaler Anonymitätsbegriffe 20

2 Verankerung von Datenschutz in der Software-Entwicklung **24**

3 Big Data – Technische Herausforderungen und Lösungen **26**

INTERVIEW MIT PROF. DR. THÜSING

Prof. Dr. Thüsing

Lieber Herr Professor Thüsing, Sie gelten als einer der führenden deutschen Juristen im Bereich des Datenschutzes. Dennoch als Einstieg einmal ganz platt gefragt: Ist Big Data auch ein Stichwort für Juristen?

Ganz bestimmt, wenn auch nur im übertragenen Sinne. *Big Data* ist kein Rechtsbegriff. Es meint die Analyse großer Datenmengen in hoher Geschwindigkeit, zumeist mit dem Ziel, diese Daten wirtschaftlich nutzbar zu machen. Die Daten sind dann oftmals aus unterschiedlichen Quellen und unterschiedlichen Formaten. Der Datenschutz setzt bei der Verarbeitung der Daten an. Alles, was mit personenbezogenen Daten getan wird, muss sich an die Spielregeln halten, die dafür geschaffen wurden. Und je mehr Daten betroffen sind, desto wichtiger ist die Einhaltung dieser Regeln.

Welche Spielregeln sind das denn?

Ich meine die verschiedenen Gesetze und Normierungen, die dafür geschaffen worden sind, den Betroffenen und seine Persönlichkeit zu schützen. Hier greifen europäische und nationale Regelungen Hand in Hand und erst eine Gesamtschau der verschiedenen Regelungsebenen kann die Frage beantworten, welcher Umgang mit Daten zulässig ist und welcher nicht. Die Datenschutz-Grundverordnung ist eine europäische, ab Mai 2018 unmittelbar geltende Regelung, zusätzlich enthält das Bundesdatenschutzgesetz ergänzende Vorschriften. Beide zusammen – und noch einige Spezialgesetze – sind bei Big Data zu beachten.

Bevor es allzu juristisch wird: Welches Ziel haben alle diese Regelungen?

Mit jedem Fortschritt der Digitalisierung wächst die Bedeutung des Datenschutzes. Lassen Sie mich etwas weiter ausholen: Person und Persönlichkeit haben eine gemeinsame etymologische Wurzel: *personare*. Das Durchtönen durch die Maske hindurch, die der Schauspieler der Antike zum besseren Ausdruck des durch ihn verkörperten Charakters trug. Es ist ein plastisches, einfach vermittelbares Bild: Die Individualität des Einzelnen hinter aller Fassade macht die Person aus, der mitunter verborgene Kern, der sich dem Blick des Gegenübers entzieht. Erst der unverstellte, maskenlose Blick erfasst die Person; wer diesen Blick verhindert, schützt seine Person vor dem Zugriff anderer. Denn Wissen um den anderen bedeutet stets auch Teilhabe am anderen. Die Wahrnehmung durch Dritte verändert die Person, ist kein Akt neutralen Erkennens, sondern performativer Gestaltung. Die Person formt und entwickelt sich nicht aus sich selber heraus, sondern im Dialog mit dem anderen; sie ist nicht statisch autonom, sondern dynamisch reflexiv. Wir steigen niemals in denselben Fluss und sind in der Zeit auch nicht unveränderte Person. *No man is an island*, und auch die Person kann nur erfasst und verstanden werden in dem sozialen Beziehungsgeflecht, zu dem sie fähig ist und das sie gebildet hat. Dieser Prozess ist ein anderer, je nachdem, wie weit die Person dem anderen offenbart wurde. In Offenheit und Vertrauen wächst eine Beziehung anders als in Anonymität und Skepsis. Wissen um den anderen bedeutet Zuordnung zum anderen. „Ich habe dich bei deinem Namen gerufen, du bist mein." – das biblische Wort ist übertragbar. Je mehr über den anderen offenbar wird, desto mehr ist seine Freiheit eingeschränkt. Wissen über den anderen ermöglicht den Kontakt, fehlendes Wissen beschränkt ihn.

Es geht also um Schutz der Persönlichkeit und letztlich auch um den Schutz der persönlichen Freiheit?

Genau! Es hat einige Zeit gedauert, bis dieser Zusammenhang verfassungsrechtliche Verankerung gefunden hat. Das ursprünglich durch die zivilrechtliche Rechtsprechung entwickelte allgemeine Persönlichkeitsrecht ist mittlerweile auch im Verfassungsrecht der zentrale Leitgedanke zur Abwehr von diversen Formen der Beeinträchtigung der Privatsphäre, die sich keinem anderen spezifischen Freiheitsrecht zuordnen lassen. Es ergänzt – in den Worten des Bundesverfassungsgerichts – „als unbenanntes Freiheitsrecht die speziellen (‚benannten') Freiheitsrechte" und schützt allgemein die „engere persönliche Lebenssphäre und die Erhaltung ihrer Grundbedingungen […], die sich durch die traditionellen konkreten Freiheitsgarantien nicht abschließend erfassen lassen".

Die für das Datenschutzrecht wichtigste Ausprägung des allgemeinen Persönlichkeitsrechts ist das Recht auf informationelle Selbstbestimmung, welches durch das grundlegende Volkszählungsurteil des Bundesverfassungsgerichts entwickelt wurde. Dieses Recht entspricht – in den Worten des Gerichts – am ehesten einem „Grundrecht auf Datenschutz", denn es schützt ganz allgemein „die Befugnis des Einzelnen, grundsätzlich selbst über die Preisgabe und Verwendung seiner persönlichen Daten zu bestimmen". Die freie Entfaltung der Persönlichkeit setzt daher den Schutz des Einzelnen gegen unbegrenzte Erhebung, Speicherung, Verwendung und Weitergabe seiner persönlichen Daten voraus. Jeder Bürger müsse grundsätzlich darüber verfügen können, „wer was wann und bei welcher Gelegenheit" über ihn weiß.

INTERVIEW MIT PROF. DR. THÜSING

Aber was heißt das dann konkret?

Big Data hat die beschriebenen Fragestellungen mit ganz neuer Dringlichkeit formuliert. Es ist die schier unvorstellbare Menge an Daten, die hier verarbeitet wird und die zukünftig noch wachsen wird. Wir leben heute in Zeiten des partiellen Datennudismus. Viele Menschen sind sich schon heute nicht bewusst, welche sensiblen Daten sie Dritten zur Verarbeitung geben. Doch nicht nur das: Neue technische Möglichkeiten – wie die automatisierte Auswertung der Daten durch eine künstliche Intelligenz – machen eine schier endlose Verknüpfung von Daten möglich und werden in Zukunft zunehmen. Diese beiden Effekte begründen die Gefahr, dass der Mensch in einer Weise „verdatet" wird, die er nicht erwartet und die er nicht will.

Das klingt immer noch sehr abstrakt. Können Sie konkreter werden?

Big-Data-Anwendungen leben von personenbezogenen Daten. Das können Vertragsdaten zu gekauften Produkten und Dienstleistungen sein. Das können aber genauso Nutzungsdaten von Apps und Daten aus sozialen Netzwerken sein bis hin zu Standortdaten von Handys oder Sensordaten von Maschinen. Hier geht es nicht nur so sehr allein um die technische Seite des Datenschutzes, sondern auch um die Grenzen des technisch Möglichen. Ich kann vielleicht einige Stichworte geben. Ein gelungener Umgang mit Big Data setzt eine Infrastruktur voraus, die gegen den unberechtigten Zugriff Dritter sichert – wir sprechen vom technischen Datenschutz, aber ebenso von Grenzen des Umgangs mit Daten, die über das Erforderliche oder vom Betroffenen Bewilligte hinausgehen. Es geht darum, mit Daten so sparsam wie möglich umzugehen, und dort, wo der Gebrauch personenbezogener Daten nicht erforderlich ist für eine bestimmte Big-Data-Anwendung, diese zu vermeiden.

Mit Datensparsamkeit und Datenvermeidung haben Sie jetzt schon einige Stichworte gegeben, die auch dem Nichtjuristen geläufig sind. Wie kann dies bei Big Data konkret aussehen?

Datensparsamkeit heißt, man soll Daten möglichst vermeiden, wo sie nicht erforderlich sind. Wenn Daten einer Big-Data-Lösung keine Personennamen enthalten, so können sie doch aufgrund anderer Umstände den Bezug zu einer Einzelperson ermöglichen und dann sind wir mittendrin im Datenschutz – denn dann handelt es sich um personenbezogene Daten. Wo also Big-Data-Lösungen diesen Bezug ausschließen können, sind sie zu bevorzugen. Auf der anderen Seite gilt Ähnliches für Daten bei Werbezwecken. Das Gesetz verlangt hier grundsätzlich die Einwilligung des Betroffenen, von einigen Ausnahmen abgesehen. Der Gesetzgeber ist hier recht streng: Werbung meint so ziemlich alles, womit man seinen Absatz fördern will und sich im Markt bekannter machen will. Hinzu kommen besondere Regelungen für den Umgang mit besonderen Daten. Überall wo es um Online-Daten geht, kann das Telemediengesetz eingreifen. Dieses verpflichtet die Unternehmen, den Betroffenen vorab zu informieren und ihm die Möglichkeit einzuräumen, der Verarbeitung seiner Daten zu widersprechen. Und auch wenn er einwilligt, muss diese Einwilligung nach bestimmten formalen Voraussetzungen erfolgen.

Nicht jede Einwilligung ist wirksam?

Nein, nein. Nur eine Einwilligung, die wirklich als Ausdruck eines freien Willens auf informierter Grundlage besteht. Nicht jedes eilig angeklickte „Meinetwegen" reicht, sondern nur die bewusste Entscheidung in einem genau umrissenen Rahmen der Verarbeitung. Versteckt in AGB ist sie regelmäßig unwirksam und auch muss sie in aller Regel schriftlich erfolgen. Es ist richtig, dass der Datenschutz nicht durchgesetzt werden muss gegen den Willen desjenigen, der durch ihn geschützt sein soll. Doch nur wenn der Grundrechtsverzicht wirklich eine bewusste Entscheidung ist, ist er legitime Grundrechtsausübung. Oftmals wird gerade das falsch gemacht. Der bloße Medienbruch reicht nicht als Rechtfertigung dafür, ausnahmsweise auf die Schriftlichkeit zu verzichten, und das neue europäische Datenschutzrecht macht es spätestens ab Mai 2018 klar, dass auch die Einwilligung in eine Datenverarbeitung nicht zur Voraussetzung des Vertragsschlusses gemacht werden darf.

Sie sprechen neue europäische Regelungen an. Was hat es damit auf sich?

Ab dem 25. Mai nächsten Jahres gilt die Datenschutz-Grundverordnung: eine europäische Regelung, die unmittelbar in das deutsche Recht hinein wirkt. Der Bundestag hat eine neue Fassung des Bundesdatenschutzgesetzes jüngst verabschiedet, die hierauf reagieren soll und eine systemkonforme Einbettung in den bestehenden Rechtsrahmen ermöglichen soll. Gerade aber der Grundsatz der Zweckbindung etwa wird durch das neue Recht noch einmal deutlich stärker als bislang betont. Daten dürfen also grundsätzlich nur mit dem Zweck verarbeitet werden,

zu dem sie erhoben wurden. Das ist bereits durch das erwähnte Volkszählungsurteil des Bundesverfassungsgerichts von 1983 hervorgehoben worden. Das Gericht betonte damals schon, dass der Verwendungszweck „bereichsspezifisch und präzise" bestimmt werden müsse. Viele sprechen beim Bindungsgrundsatz von einem Kernstück datenschutzrechtlicher Regelungen.

Nun heißt es im europäischen Recht, dass eine Verarbeitung nicht mit dem bei Erhebung festgelegten Zweck unvereinbar sein darf. Das ist im Einzelnen schwer zu bestimmen, wo hier die Grenzen liegen. Wichtig ist aber: Ist ein Zweck einmal für die Erhebung und Verwendung von Daten gesetzt, so ist dieser bei später folgenden Verwendungen grundsätzlich einzuhalten. Jede Zweckänderung muss gerechtfertigt werden und ob diese Rechtfertigung besteht, richtet sich eben nach der Einwilligung des Betroffenen oder aber nach den eng umrissenen Ausnahmetatbeständen des Bundesdatenschutzgesetzes oder der Datenschutzgrundverordnung, die hier regelmäßig an der Erforderlichkeit anknüpfen. Oftmals wird erforderlich sein, zu einer Pseudonymisierung und Anonymisierung zu kommen, um den Personenbezug zu verhindern. Wo das nicht möglich ist, ist eine detaillierte gesetzliche Analyse notwendig, um die präzisen Grenzen zu ermitteln. Und diese Grenzen sind enger als ehemals.

Kann das zu Wettbewerbsnachteilen deutscher Unternehmen führen?

Ja, sicherlich. In den USA haben wir beileibe nicht ein Datenschutzrecht vergleichbar dem europäischen Standard und eben deswegen ringen ja auch die Europäische Union und die USA um die richtigen Regelungen zum Datenexport in die USA. Hier wurde in mühevoller Kleinarbeit das sogenannte Privacy Shield verhandelt und vereinbart, das einen Schutz auch von Daten aus europäischen Quellen in den USA realisieren soll.

Das US-amerikanische Recht selber hat aber ein solches Datenschutzniveau nicht und hat nur im richterrechtlichen Ansatz einige dünne Schutzprinzipien aus den Gedanken des *common law of privacy* herausgearbeitet. Big Data ist in den USA daher einfacher zu verarbeiten als in Europa und daher kommt vielleicht auch die datenschutzrechlich sportliche Einstellung von Unternehmen wie Google und Facebook, die hier mit ihren AGB oftmals die Grenze des Möglichen austesten wollen.

Wird sich das in Zukunft ändern?

Nun ja, Digitalisierung heißt auch Internationalisierung und dort, wo Google und Facebook im europäischen Markt handeln, dort müssen Sie sich auch an das europäische Recht halten. Was in den USA und in anderen Ländern der Welt möglich sein mag, ist es in Europa nicht notwendigerweise. Hier kann dem Handel mit der Datenverarbeitung amerikanischer Unternehmen durchaus Einhalt geboten werden. Der Europäische Gerichtshof hat sich gerade in den vergangenen Jahren insbesondere mit Google beschäftigt und das neue Recht hat vor allem eine Neuerung: Sehr viel härtere Sanktionen.

Bislang war in Deutschland ein Datenverstoß mit höchstens 300.000 Euro Buße versehen. Zukünftig können es bis zu 4 % des weltweiten Jahresumsatzes sein, den ein Datenfehler kostet. Stellen Sie sich das vor: Bei Dax-Konzernen kann das bis in die Milliarden gehen. Wenn man bedenkt, dass die Umsatzrendite oftmals unter 4 % liegt, könnten theoretisch mehrere Jahresgewinne abgeschöpft werden. Ein wenig salopp formuliert: Das Geld, was die Amerikaner momentan von VW für die Abgasskandale und von der Deutschen Bank für die Aufsichtsfehler einfordern, könnte Europa künftig von amerikanischen Unternehmen auf Grundlage des Datenschutzes zurückfordern.

Bei solch unglaublich hohen Bußgeldern: Wird sich dadurch im Datenschutz auch inhaltlich etwas bewegen?

Ich bin mir sicher: Datenschutz wird zukünftig ernster genommen werden, weil die Risiken mit dem fehlerhaften Umgang von Daten größer werden und weil das Bewusstsein für den richtigen Umgang mit Daten vielleicht doch stärker geworden ist. Unternehmen, die Datenschutz ernst nehmen, haben damit auch beim Kunden einen Vorteil gegenüber Unternehmen, denen diese Expertise fehlt. Nach dem Datenskandal der Deutschen Bahn musste der Vorstandsvorsitzende gehen und so zeigen auch nicht-rechtliche Sanktionen, wie etwa hier die Schädigung des Rufs in der Öffentlichkeit, dass Unternehmen mit Daten sensibel agieren müssen – auch und gerade bei Big Data, wo die Gefährdungen besonders hoch sind.

Herr Professor Thüsing, wir danken Ihnen für dieses Gespräch.

1 METHODEN ZUR DATENSCHUTZKONFORMEN AUSWERTUNG PERSONENBEZOGENER INFORMATIONEN

Wenn von „moderner Kryptographie" die Rede ist, bezieht sich das oft auf sogenannte „asymmetrische Verschlüsselung". Anders als bei symmetrischen Verfahren, bei denen derselbe geheime Schlüssel zum Ver- und Entschlüsseln verwendet wird, erlauben asymmtrische Verfahren die bedenkenlose Weitergabe eines öffentlichen Verschlüsselungsschlüssels. Mit diesem kann jeder verschlüsselte Nachrichten erstellen, nicht jedoch den Inhalt ihm unbekannter Chiffrate lesen. Der zugehörige private Entschlüsselungsschlüssel muss dabei natürlich vom Schlüsseleigner weiter unter Verschluss gehalten werden. Unter Ausnutzung derselben mathematischen Prinzipien (aber auf umgekehrtem Weg) arbeiten digitale Signaturen: Nur der Eigner des privaten Schlüssels kann eine gültige Signatur erstellen, aber jedermann kann mittels des entsprechenden öffentlichen Schlüssels die Gültigkeit einer Signatur überprüfen.

Die Entwicklung und Etablierung asymmetrischer Verschlüsselungsverfahren und digitaler Signaturen kann in vielerlei Hinsicht als einer der wichtigsten Meilensteine in der Geschichte der Computersicherheit angesehen werden. Der Zenit kryptographischer Möglichkeiten ist damit aber längst nicht erreicht. Methoden der sogenannten „sicheren Mehrparteienberechnung" gehen weit über das hinaus, was allein mit Verschlüsselungs- und Signaturprotokollen erreichbar wäre. Zumindest in der Theorie sind allgemeine Datenverarbeitungsmethoden, die den eigentlichen Dateninhalt selbst vor den verarbeitenden Instanzen absolut geheim halten, schon seit Längerem bekannt.

Durch jüngere Forschungsdurchbrüche und den allgemeinen technologischen Fortschritt bei der elektronischen Datenverarbeitungsgeschwindigkeit rücken diese Methoden immer mehr in den Bereich des wirtschaftlich Machbaren. Gleichzeitig wächst auch der Bedarf an solchen Datenverarbeitungsmethoden mit besonders strikten Datenschutzgarantien, um die immer weitläufiger verfügbaren personenbezogenen Daten für statistische Analysen nutzen zu können.

In den nachfolgenden Unterkapiteln werden entsprechende Lösungsansätze aus drei verschiedenen Perspektiven vorgestellt:

- Das Kapitel **Grenzen klassischer IT-Security und darüber hinausgehende Mittel der modernen Kryptographie** bietet eine allgemeine Einführung in die algorithmischen Werkzeuge aus IT-Security und Kryptographie, auf denen aufbauend („bottom-up") eine umfassende Datenauswertung unter Einhaltung strengster Datenschutzanforderungen realisiert werden kann. Dabei wird auch auf den jeweiligen Rechenaufwand und Optimierungspotentiale eingegangen.

- Orthogonal dazu finden sich im Kapitel **Design-Prinzipien für datenschutzkritische Applikationen** die zu erzielenden Anwendungseigenschaften und exemplarisch daraus abgeleitete Maßnahmen („top-down"). Die Anwendbarkeit der dort beschriebenen Prinzipien ist nicht auf Software-Projekte beschränkt, in denen spezielle oder besonders neuartige Kryptographiemethoden zum Einsatz kommen. Vielmehr ist für lückenlose Sicherheit gerade bei Lösungen, die rein auf klassischen IT-Security-Maßnahmen basieren, ein umfassendes und gut abgestimmtes Gesamtkonzept erforderlich.

- Das Kapitel **Notwendigkeit, Stärken und Schwächen formaler Anonymitätsbegriffe** behandelt schließlich die Frage, unter welchen Voraussetzungen eine statistische Auswertung personenbezogener Daten überhaupt als hinreichend anonym gelten kann, um veröffentlicht und/oder anderweitig verwendet werden zu dürfen – denn selbst ein perfekt gegen alle Arten von Angriffen abgesicherter Berechnungsprozess liefert per se noch keinerlei Garantie, dass das Berechnungsergebnis nicht trotzdem datenschutzkritischen Personenbezug enthält.

1.1 GRENZEN KLASSISCHER IT-SECURITY UND DARÜBER HINAUSGEHENDE MITTEL DER MODERNEN KRYPTOGRAPHIE

» Klassische IT-Security-Ansätze stoßen an ihre Grenzen, wenn überhaupt keine Entität mehr Zugriff auf die zu schützenden Daten haben darf. Moderne Kryptographie bietet mit Mitteln der sogenannten „sicheren Mehrparteienberechnung" einen möglichen Ausweg, diese strikte Anforderung zu erfüllen und grundsätzlich jede beliebige statistische Berechnung datenschutzkonform zu realisieren. Die generischen Verfahren sind allerdings meist zu aufwändig für den Praxiseinsatz. Mit anwendungsspezifischen Spezialverfahren kann ein wirtschaftliches Aufwandsniveau erreicht werden, aber dafür ist umfassende Kryptographie- und Datenschutzexpertise notwendig.

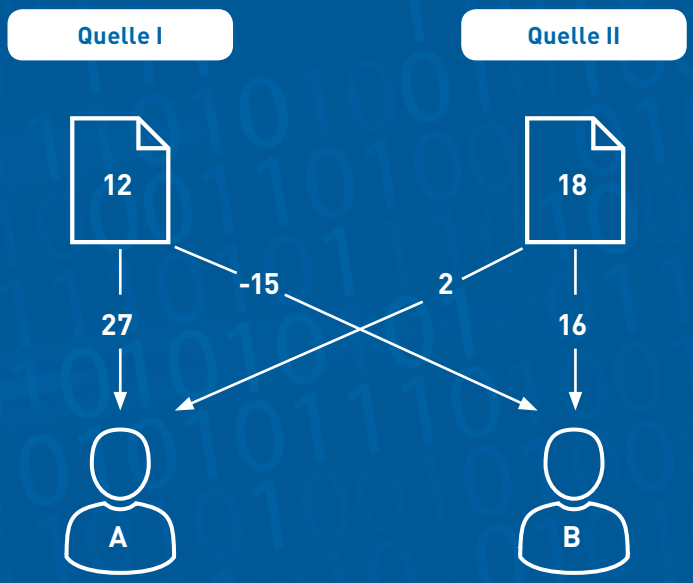
IT-Security hat generell zum Ziel, Systeminterna gegen unerwünschte äußere Einflüsse abzuschotten und unbefugte Preisgabe von Informationen zu unterbinden. Klassisch wird dabei das Systeminnere als vertrauenswürdig angesehen. Zugriffe von außen sind dagegen nur über dedizierte Schnittstellen möglich, an denen das System zunächst überprüft, ob eine Zugriffsaktion gemäß den geltenden Access-Control-Regeln zulässig ist, und im Erfolgsfall die gewünschte Operation ausführt. Dahinter liegt eine gewisse Schwarzweiß-Sicht, die die Welt in legitimierte Systembenutzer und abzuwehrende Angreifer unterteilt. Prinzipiell können die legitimen Systembenutzer dabei beliebig feingranular in Untergruppen mit unterschiedlichen Rollen und jeweils individuellen Zugriffsrechten unterteilt werden. Dennoch gilt für jede dieser Gruppen, dass beispielsweise eine Datei entweder vollumfänglich gelesen werden kann oder aber der Zugriff komplett verweigert wird. Soll ein User (in seiner aktuellen Rolle) lediglich Teilzugriff auf gewisse Daten erhalten, so ist dies nur wieder über einen Systemprozess realisierbar, welcher selbst Vollzugriff auf die Daten hat und für den User die entsprechend eingeschränkte Datenansicht bereitstellt.

Dieses Paradigma stößt an seine Grenzen, sobald aus Datenschutzgründen eigentlich gar keine Entität mehr existieren darf, die Vollzugriff auf die Daten hat. Das ist unter anderem dann der Fall, wenn Statistiken auf Basis personenbezogener Nutzer- oder Kundendaten kommerziell genutzt werden sollen. Weitere Anwendungsfälle entstehen zum Beispiel bei präventiver Systemüberwachung („Vorratsdatenspeicherung"), wo nur bei kriminellem Verhalten eines Anwenders der Rückgriff auf entsprechende Überwachungsdaten möglich sein darf. Auf konventionellem Weg ist das hierfür erforderliche Schutzniveau nur über technische Selbstversiegelung zu erreichen: Einmal im Produktivbetrieb gestartet, sperrt das System alle Wartungszugänge, über die sensitive Daten abgegriffen werden könnten (das betrifft insbesondere auch den physischen Zutritt zu den Server-Räumen), und kann nur durch Rücksetzung in den datenfreien Ursprungszustand für Updates, Upgrades oder Ähnliches wieder zugänglich gemacht werden.

Eine Alternative oder vielmehr Ergänzung zu den klassischen IT-Security-Mechanismen bietet die moderne Kryptographie mit Mitteln der sogenannten „sicheren Mehrparteienberechnung" (englisch „secure multi-party computation"). Dabei führen mehrere Parteien gemeinsam wertschöpfende Datenanalysen so durch, dass jede einzelne Partei zu jedem beliebigen Zeitpunkt ausschließlich zusammenhanglosen Zufall sieht. Methoden der klassischen IT-Security können höchstens die jeweilige lokale Datenansicht der beteiligten Parteien einschränken (indem beispielsweise jedes einzelne Attribut eines Datensatzes mit einem anderen Schlüssel verschlüsselt wird und jede an der Datenanalyse beteiligte Partei nur für eine gewisse Teilmenge davon die zugehörigen Schlüssel besitzt). Der Privatsphäreneingriff wird dadurch zwar abgeschwächt, der faktische Personenbezug der verarbeiteten Daten in letzter Konsequenz aber nie vollständig aufgelöst. Sichere Mehrparteienberechnung bietet dagegen die absolute Garantie, dass tatsächlich niemand mehr Zugriff auf irgendwie geartete Information mit Personenbezug hat. Es existiert eine Reihe unterschiedlicher Techniken, dies zu erreichen.

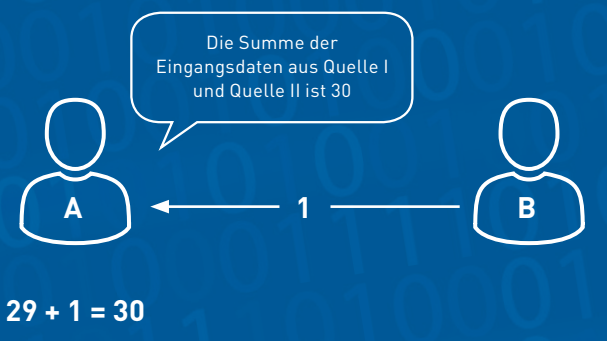
Abbildung: Beispielprotokoll basierend auf Secret Sharing für eine einfache Datenanalyse, ohne dass die beteiligten Parteien die Eingangsdaten sehen.

SECRET SHARING FÜR EINE EINACHE DATANANALYSE



Schritt 1
Die Eingangsdaten werden als Summen von Zufallszahlen dargestellt, z.B. 18 = 2 + 16. Die Shares 27, -15, 2 und 16 werden so auf die Parteien A und B verteilt, dass keine Partei die Ursprungsdaten rekonstruieren kann.

Schritt 2
Jede Partei berechnet für sich lokal die Summe der erhaltenen Shares. Die berechneten Summen sind immer noch Zufallszahlen, die nichts über die Ursprungsdaten aus Quelle I und Quelle II verraten.



Schritt 3
Aus den lokal berechneten Teilsummen können Partei A und B gemeinsam die korrekte Gesamtsumme berechnen, ohne je die ursprünglichen Eingangsdaten gesehen zu haben.

SECRET SHARING

Eine der simpelsten Formen eines Secret Sharing ist, einen Zahlenwert als Summe zweier jeweils für sich vollkommen zufällig aussehender Summanden darzustellen und jeden Summanden („Share") an eine andere Partei zu geben. Hat man auf diese Weise viele Zahlenwerte auf dieselben zwei Parteien verteilt, können diese zum Beispiel gemeinsam die Gesamtsumme (und damit auch den Durchschnitt) darüber berechnen, ohne dass irgendeine der Parteien jemals einen der ursprünglichen Zahlenwerte sieht: Jede Partei addiert zunächst lokal alle ihre Shares zusammen, die erste Partei sendet ihr Ergebnis an die zweite Partei und diese kann letztendlich aus den beiden Teilsummen die Gesamtsumme bilden.

Komplexere Secret-Sharing-Verfahren erlauben es (zum Beispiel aufbauend auf Reed-Solomon-Codes), Daten auf eine beliebig große Anzahl an Parteien aufzuteilen und beliebige Funktionen so auf den Shares auszuwerten, dass jede einzelne Partei nur zu sammenhanglose Zufallsdaten sieht. Einzige Randbedingung ist, dass sich nie die Hälfte oder mehr der beteiligten Parteien bösartig zusammenschließt, um die Ursprungsdaten aus den Shares zu rekonstruieren. Unter der verschärften Randbedingung, dass weniger als ein Drittel der Parteien korrumpiert ist, garantiert das System sogar Robustheit gegen jegliche Angriffe auf die Korrektheit der Berechnungsergebnisse.

Man beachte, dass aufgrund der Forderung nach einer vertrauenswürdigen Mehrheit solch eine Lösung im Zwei-Parteien-Fall keinen Mehrwert bietet, da beide Parteien absolut unkorrumpierbar sein müssten. Es ist zwar wie im Eingangsbeispiel immer möglich, Information so auf zwei Parteien aufzuteilen, dass jede einzelne Partei lediglich zusammenhanglosen Zufall sieht. Allerdings können auf solchen Shares nur sehr eingeschränkte Arten von Berechnungen durchgeführt werden. Rein auf Secret Sharing basierte Verfahren haben den Vorteil, dass sie informationstheoretisch sicher sind und nicht wie Verschlüsselung auf der angenommenen Unlösbarkeit bestimmter Berechnungsprobleme beruhen. Im Gegenzug erfordern allgemeine Berechnungen auf den Shares aber einen hohen Kommunikationsaufwand zwischen den beteiligten Parteien.

GARBLED CIRCUITS

Garbled Circuits sind eine Methode, die auf klassischer Verschlüsselung basiert und für kryptographisch sichere Berechnungen im Zwei-Parteien-Fall Anwendung findet. Sie sind konzeptuell wesentlich komplexer als Secret Sharings, aber es lassen sich damit die entsprechenden Einschränkungen auf Szenarien mit vertrauenswürdiger Mehrheit vermeiden. Ein zusätzlicher Vorteil ist eine vergleichsweise geringe Anzahl der erforderlichen Kommunikationsrunden.

Im Wesentlichen wird beim Garbled-Circuits-Ansatz die auszuwertende Funktion als logischer Schaltkreis in einer speziellen verschlüsselten Form dargestellt. Der Schaltkreis lässt sich in dieser speziellen Form auswerten und so das gewünschte Funktionsergebnis ohne Klartextzugriff auf die Eingangsdaten berechnen. Dabei ist im Wesentlichen pro Bit-Operation eine Ver- bzw. Entschlüsselung mit einem (im Vorfeld beliebig wählbaren) symmetrischen Verfahren nötig, was auf den ersten Blick sehr rechenaufwändig erscheinen mag. Allerdings sind mittlerweile verschiedene Optimierungen bekannt. So lässt sich unter anderem mittels eigens dafür optimierter Schaltkreis-Layout-Verfahren die Anzahl der insgesamt notwendigen Ver- und Entschlüsselungen minimieren. Ferner verfügen viele gängige Prozessoren über Hardware-Unterstützung für symmetrische Ver- und Entschlüsselung (mit AES), was die Performance gegenüber einer reinen Software-Implementierung noch einmal deutlich verbessert.

HOMOMORPHE VERSCHLÜSSELUNG

Homomorphe Verschlüsselung bietet die Möglichkeit, auf verschlüsselten Werten zu rechnen, ohne die enthaltenen Klartexte oder gar den Entschlüsselungsschlüssel zu kennen. In der Theorie sind Verfahren bekannt, die beliebige Operationen auf den Chiffraten zulassen. Damit ließen sich Lösungen für umfassende statistische Personendatenauswertungen bei bestmöglichem Datenschutz entwickeln, wenn diese sogenannten „vollhomomorphen" Verfahren nicht aufgrund ihres exorbitanten Rechenaufwands weit von der praktischen Einsetzbarkeit entfernt wären.

Teilhomomorphe Verfahren dagegen, welche nur bestimmte Arten von Operationen auf den Chiffraten zulassen (zum Beispiel ausschließlich Addition der enthaltenen Werte), bewegen sich aufwandsmäßig in ähnlichen Größenordnungen wie gewöhnliche Public- Key-Verschlüsselung und sind damit durchaus in der Praxis einsetzbar. Der praktische Nutzen vor allem im Big-Data-Umfeld ist aber eher begrenzt, da Public-Key-Verschlüsselung auf großen Datenmengen nach wie vor sehr rechenintensiv ist.

OBLIVIOUS TRANSFER

Mittels teilhomomorpher Verschlüsselung bzw. verwandter mathematischer Techniken aus der Public-Key-Kryptographie lassen sich mit vertrebarem Aufwand einfache Grundbausteine sicherer Mehrparteienberechnung realisieren. Ein besonders bedeutender Grundbaustein heißt „Oblivious Transfer" und lässt eine Partei (den „Sender") eine beliebige Ein-Bit-Funktion spezifizieren, die eine andere Partei (der „Empfänger") dann genau ein Mal für eine beliebige Ein-Bit-Eingabe auswerten kann. Die essenzielle Eigenschaft von Oblivious Transfer ist dabei, dass keine der Parteien weitergehende Informationen erhält: Vor dem Sender bleibt die Eingabewahl des Empfängers vollständig verborgen, während gleichzeitig der Empfänger außer dem einmalig beobachteten Ein-Ausgabe-Verhalten nichts über die vom Sender gewählte Funktionsspezifikation erfährt. Aufbauend auf Oblivious Transfer lassen sich beliebig komplexe sichere Mehrparteienberechnungen zusammensetzen – im weitesten Sinne analog dazu, wie sich beliebig komplexe logische Schaltkreise aus einfachen Bit-Operationen (UND-, ODER-, NICHT-Gatter) zusammensetzen lassen.

Die Auswertung einer einzelnen Oblivious-Transfer-Instanz ist so aufwändig wie herkömmliche Public-Key-Verschlüsselung und damit eigentlich nicht Big-Data-tauglich, es existieren aber Amortisierungsmethoden (analog zu hybrider Verschlüsselung), mit denen der Gesamtaufwand für hinreichend viele parallele Oblivious-Transfer-Instanzen im Wesentlichen auf das Niveau von symmetrischer Verschlüsselung gesenkt werden kann. Da dieser Effekt schon bei recht geringen Datenmengen voll zum Tragen kommt, ist es legitim, Oblivious Transfer im Allgemeinen als ähnlich aufwändig wie symmetrische Verschlüsselung zu betrachten.

Generell kann mittels sicherer Mehrparteienberechnung ein Schutzniveau äquivalent zur Datenspeicherung und -prozessierung in einer perfekt vertrauenswürdigen Entität erreicht werden, ohne dass eben jene tatsächlich physisch extistieren muss. Vielmehr reicht für die Gesamtsicherheit des Systems aus, wenn irgendeine

beliebige Einzelkomponente unkorrumpiert bleibt – selbst wenn alle anderen Systembereiche kollaborativ versuchen, die Datensicherheit zu unterminieren. Übersetzt auf mögliche Privatsphärenverletzungen durch interne Angreifer bedeutet das, dass die Vertraulichkeit der betroffenen personenbezogenen Daten gewahrt bleibt, solange nicht alle Systembetreiber bösartig zusammenarbeiten.

Der Preis für dieses hohe Maß an Sicherheit ist ein erhöhter Rechen- und Kommunikationsaufwand. Secret-Sharing-Arithmetik ist in erster Näherung ähnlich aufwändig wie normale Grundrechenoperationen. Allerdings erfolgen die Berechnungen in einer speziellen mathematischen Struktur (einem „endlichen Körper") die nicht immer ohne Weiteres auf die gewünschte Datenanalyse-Anwendung übertragbar ist. Der resultierende Overhead bei der Rechenzeit kann mehr als eine Größenordnung betragen. Symmetrische Verschlüsselung und damit auch Garbled Circuits und Oblivious Transfer sind um mehrere Größenordnungen aufwändiger als Secret-Sharing-Arithmetik. Das genannte Problem, dass Berechnungen nicht in einer direkt auf die Anwendung übertragbaren mathematischen Struktur erfolgen, besteht dabei ebenfalls und treibt den Gesamtaufwand weiter in die Höhe. Public-Key-Verschlüsselung und damit auch homomorphe Verfahren sind noch einmal um mehrere Größenordnungen aufwändiger als symmetrische Verschlüsselung. Hinzu kommt bei sämtlichen genannten Verfahren, dass alle Berechnungsoperationen in irgendeiner Form von jeder einzelnen beteiligten Partei durchgeführt werden müssen – mit der Anzahl der Parteien, auf die die Daten aufgeteilt sind, steigt also neben dem Sicherheitsniveau auch der Gesamtrechenaufwand.

Gerade im Big-Data-Umfeld, wo die Kosten für jede zusätzliche Rechenoperation durch die enorme Datenmenge stark vervielfacht werden, sind solche Aufwandstreiber meist ein striktes No-Go. In der wirtschaftlichen Realität wird deshalb oft eine Mischung aus klassischen IT-Security-Methoden und Ansätzen aus dem Gebiet der sicheren Mehrparteienberechnung der einzig gangbare Weg sein. In der Regel besteht auch gar nicht die Anforderung, jedes einzelne Bit an irgendwie personenbeziehbarer Information perfekt geheim zu halten – es muss nur sichergestellt sein, dass niemand ohne hohen technischen Aufwand und starke kriminelle Energie Zugriff auf Daten mit Missbrauchspotential hat. Mit den richtigen Kompromissen lassen sich durchaus Lösungen finden, die bei vertretbarem Aufwand sowohl strengen Datenschutzanforderungen als auch umfassenden Datenverwertungsinteressen gerecht werden.

Die Vergangenheit hat allerdings auch gezeigt, dass Risiken und Missbrauchspotentiale gerne drastisch unterschätzt werden und die eingebrachte Expertise und Erfahrung bei der entsprechenden Gefahrenanalyse eigentlich nicht groß genug sein können. Hinzu kommt, dass Analyseergebnisse schon allein durch den allgemeinen technologischen Fortschritt und/oder Änderungen im Einsatzumfeld einer Software ungültig werden können. Insbesondere dürfen einmal getroffene Einschätzungen und Entscheidungen nicht ad acta gelegt werden, sondern müssen fortwährend kritisch hinterfragt werden.

Zusätzliches Optimierungspotential beim Entwickeln einer konkreten Software-Lösung ergibt sich auch daraus, dass die genannten kryptographischen Methoden generische Konstruktionen sind, mit denen sich prinzipiell jede beliebige Berechnung datenschutzkonform realisieren lässt. Dieser grundsätzlichen theoretischen Machbarkeit steht in jedem praktischen Einzelfall neu die Frage gegenüber, mit welcher speziell auf die Projektanforderungen zugeschnittenen Methodik der Ressourcenaufwand auf ein wirtschaftlicheres Maß reduziert werden kann.

Ein vielversprechendes Vorgehen besteht oftmals darin, aufwändige generische Verfahren nur zum Erzeugen eines initialen sicheren Setups zu benutzen und darauf aufbauend die eigentliche Datenverarbeitung mit sehr viel effizienteren Spezialverfahren durchzuführen. Idealerweise können kryptographisch sichere, aber in ihrer Ursprungsform zu teure Verfahren über solch einen Armortisierungs- bzw. Bootstrapping-Ansatz Big-Data-tauglich gemacht werden, ohne dass dafür Abstriche beim Datenschutz notwendig wären. Ähnlich wie die Risikoanalyse birgt allerdings auch das Design anwendungsspezifischer Methoden und Protokolle grundsätzlich ein hohes Potential für subtile, aber datenschutzkritische Fehler.

1.2 DESIGN-PRINZIPIEN FÜR DATENSCHUTZKRITISCHE APPLIKATIONEN

» Design-Prinzipien unterstützen einen nachhaltigen Entwicklungsprozess, bei dem die wesentlichen Projektziele im Mittelpunkt stehen und das Gesamtkonzept „aus einem Guss" bleibt. Außerdem sollte die Systemkomplexität möglichst gering gehalten werden, um die Gefahr für Implementierungs- und Anwendungsfehler zu minimieren. Das ist besonders wichtig bei Themen wie Privatsphärenschutz und/oder Big Data, die bereits inhärent eine hohe fachliche Komplexität mit sich bringen. Wir stellen eine Reihe von Prinzipien vor, die sich im Projektgeschäft beim Entwurf einer sicheren Systemarchitektur sowie entsprechender Betriebskonzepte als nützlich erwiesen haben.

Design-Prinzipien sind eine Orientierungshilfe, die dabei unterstützen soll, in Planungs- und Entwicklungsprozessen nachhaltige Entscheidungen zu treffen. Inhaltlich ist ihr Hauptzweck, die wesentlichen Projektziele in den Mittelpunkt der Entscheidungsfindung zu rücken und dafür zu sorgen, dass das Gesamtkonzept „aus einem Guss" bleibt. Als nützlicher Nebeneffekt können sie die Kommunikation sowohl innerhalb eines Projektteams als auch mit Drittparteien vereinfachen und effizienter gestalten, zum Beispiel wenn Systemeigenschaften gegenüber einem externen Gutachter oder einem von der Personendatenauswertung betroffenen Nutzer erklärt werden müssen.

Die nachfolgende Aufstellung von möglichen Design-Prinzipien hat sich im Projektumfeld rund um datenschutzkonforme Datensammlung und -auswertung als nützlich erwiesen. Die einzelnen Prinzipien sind dabei nicht unkorreliert, sondern greifen ineinander, überlappen und ergänzen sich gegenseitig. Sie sind als Erweiterung und nicht als Ersatz für gängige Sicherheitsrichtlinien und Qualitätsstandards in der Software-Entwicklung zu verstehen.

Viele der vorgestellten Prinzipien können und sollten nicht nur bei der Systemarchitektur Anwendung finden, sondern auch im zugehörigen Betriebskonzept – denn Erstere kann niemals sicher sein ohne entsprechende Vorkehrungen bei Letzterem. Die Anwendbarkeit ist auch nicht auf Lösungen beschränkt, bei denen über klassische IT-Security-Mechanismen hinausgehende Methoden der modernen Kryptographie zum Einsatz kommen. Vielmehr ist gerade in Szenarien, wo hauptsächlich oder sogar ausschließlich mit herkömmlichen Sicherheitsmaßnahmen gearbeitet wird, ein zielorientiertes und gut abgestimmtes Design der Sicherheitsarchitektur besonders wichtig.

Es liegt jedoch in der Natur der Sache, dass selbst strikte Befolgung der vorgestellten Prinzipien langjährige Privatsphärenschutz- und Kryptographie-Erfahrung nicht ersetzen oder gar als Ergebnis ein vollkommen datenschutzkonformes Entwicklungsprodukt garantieren kann. Es ist auch gar nicht immer möglich, alle Prinzipien gleichzeitig perfekt zu erfüllen.

VOLLSTÄNDIGES SECURITY-MODELL: DEFENCE IN DEPTH & FAIL-SAFE DEFAULTS

Zu einer vollständigen und möglichst robusten Sicherheitsarchitektur gehört, dass Sicherheitsmaßnahmen integraler Bestandteil sämtlicher Systemkomponenten sind (soweit dies sinnvoll möglich ist) und auch potentielle Fehlfunktionen durch sichere Rückfallmechanismen oder entsprechend restriktive Standardeinstellungen aufgefangen werden. Insbesondere darf die Sicherheit eines Moduls oder Systembereichs nicht darauf basieren, dass sich andere Teile des Systems oder gar menschliche Anwender immer gemäß einer gewissen Spezifikation verhalten. Vor allem User-Eingaben sind rigoros zu validieren, bevor sie funktional verarbeitet werden können. Aber auch zwischen voll automatisierten Systembereichen sind Schutzbarrieren sinnvoll, um das potentielle Schadensausmaß im Fall eines Sicherheitslecks auf den betroffenen Teilbereich zu begrenzen

(„Defence in Depth"). Dabei sollten Freigabe- und Zugriffsmechanismen grundsätzlich so aufgesetzt sein, dass initial keinerlei Datenzugriff und damit auch keine Datenschutzverletzung möglich ist („Fail-Safe Defaults"), legitime Operationen ausschließlich bei Bedarf freigeschaltet werden und für jede erneute Aktion auch die entsprechenden Zugriffsrechte erneut beantragt werden müssen.

Der Grund für solch ein restriktives Vorgehen ist zweifach:

- Auch im Fehlerfall soll schlimmstenfalls nur die erwartete Funktionalität nicht zur Verfügung stehen, es darf aber kein unberechtigter Zugriff auf sensible Daten möglich sein. Ersteres fällt im Betrieb der Anwendung sehr schnell auf und kann behoben werden. Letzteres wird in den meisten Fällen entweder gar nicht oder nur mit großem zeitlichen Verzug bemerkt.

- Der Schaden, wenn Funktionalität fälschlicherweise nicht zur Verfügung steht, ist meist reparierbar, indem die gewünschte Berechnung nach einem entsprechenden Bugfix erneut angestoßen wird. Der durch ein Datenleck verursachte Schaden ist dagegen in aller Regel unumkehrbar.

Die Anwendbarkeit dieser Prinzipien ist grundsätzlich nicht auf Access-Control-Mechanismen beschränkt. Eine eklatante Verletzung von Fail-Safe Defaults wäre etwa auch ein Modul, welches bestimmte Datenfelder in einem Datenstrom mit entsprechend verschlüsselten Werten überschreiben soll und bei dessen Ausfall (beispielsweise weil die Gültigkeit des verwendeten Schlüssels abgelaufen ist) plötzlich alle Daten im Klartext durchkommen. Sowohl Defence in Depth als auch Fail-Safe Defaults sind eigentlich Standardprinzipien sicherer Software-Entwicklung, erhalten im Kontext potentieller interner Angreifer und sensibler Personendatenverarbeitung aber noch einmal besondere Bedeutung und sind eng verzahnt mit den weiteren hier vorgestellten Design-Prinzipien.

SEPARATION OF DUTIES, NEED-TO-KNOW-PRINZIP, PRINCIPLE OF LEAST PRIVILEGE

Sämtliche Module sollten jeweils streng eingegrenzte Aufgabenbereiche haben und zwar vorzugsweise so, dass einzelne Komponenten bei zufälliger oder auch bösartig herbeigeführter Fehlfunktion möglichst geringen bis gar keinen Schaden anrichten können. Idealerweise sieht und darf jeder Systemteil nur genau das, was zu seiner Funktionserfüllung unbedingt notwendig ist. Das bedeutet sowohl Selbsteinschränkung beim Lesen von Daten, die aktuell nicht vollumfänglich gebraucht werden, als auch möglichst restriktive Einschränkung und Kontrolle von außen (zum Beispiel durch das Rechtemanagement des Betriebssystems).

Folgende Liste ist eine kleine Auswahl an Best Practices, die diesen Prinzipien folgen:

- Kein Zugriff auf Daten, die nicht tatsächlich zur aktuellen Funktionserfüllung benötigt werden.

- Datenverarbeitung nach Möglichkeit in einem verschlüsselten (oder als Secret Sharing verteilten) Format.

- Möglichst frühzeitige Datenanonymisierung und -aggregation in der Verarbeitungskette.

- Zusammenführung von Daten aus unterschiedlichen Quellen möglichst spät in der Verarbeitungskette.

- Prozesse laufen grundsätzlich mit weitestmöglich eingeschränkten Zugriffsrechten.

ALL-OVER ENCAPSULATION

Die größte Gefahr für Datenschutzverletzungen geht traditionell von dort aus, wo besonders direkter Zugriff auf die datenverarbeitenden Systeme besteht. Das ist in erster Linie bei Systemadministratoren und Operations-Personal gegeben. Es gibt viele mögliche Gründe, warum illegitim auf systeminterne Daten zugegriffen werden könnte. Das Spektrum reicht von Arglosigkeit (zum Beispiel wenn schlecht geschulte Mitarbeiter über

Social Engineering zur Datenweitergabe überredet werden können) über einfache Neugier bis hin zu kriminellen Intentionen. Auch reine Bequemlichkeit kann schon ein hinreichendes Motiv sein, wenn vorgeschriebene Arbeitsabläufe besonders umständlich sind und der Zugriff auf eigentlich verbotene Ressourcen den einfacheren Weg darstellt. Je niedriger die technischen Hürden für unberechtigte Datenzugriffe sind, desto harmlosere Anlässe reichen für Datenschutzverletzungen aus. Idealerweise sind nirgends sensible Daten überhaupt lokal vorhanden (sondern beispielsweise über Secret Sharing sicher verteilt) – aber selbst dann ist es sinnvoll, den lokalen Datenzugriff so weit wie möglich zu erschweren, um auch die Hürden für systemübergreifende Angriffe möglichst hoch zu setzen.

Konkrete Beispiele:

- Falls die vorhandene Netzwerkarchitektur eine Terminierung von Transportverschlüsselung grundsätzlich bereits am Übergang vom Extranet ins Intranet vorsieht, so muss Ende-zu-Ende-Verschlüsselung, wenn sie auch gegen interne Angreifer schützen soll, zusätzlich auf Applikationsebene erfolgen.

- Produktiv-Code darf keinerlei Wartungszugänge, Test- oder Debug-Modi haben, durch die sich Datenschutzmechanismen umgehen lassen (beispielsweise durch Deaktivierung von Verschlüsselung oder das Mitloggen von sensitiven Klartexten, Schlüsseln, Passwörtern, etc.).

- Die Verwendung von Standard-Interfaces (zum Beispiel beim Zugriff einer Anwendung auf eine Datenbank) ist zwar mit gutem Grund eine Best Practice zur Vermeidung von Fehlern – dabei ist aber grundsätzlich zu bedenken, dass auf sie auch über Third-Party-Tools zugegriffen werden kann, deren Funktionsumfang nicht derselben Kontrolle unterliegt wie der anwendungsspezifischer proprietärer Werkzeuge. Allerdings bietet das Abstellen auf eigene Formate ohnehin nur äußerst schwachen Schutz („Security by Obscurity") und sollte bestenfalls zusätzlich zu starken kryptographischen Maßnahmen Anwendung finden.

  Idealerweise werden gewollte Kompatibilitätsbrüche zu Third-Party-Tools ganz gezielt punktuell herbeigeführt, zum Beispiel indem Datenbankzugangsdaten nur in einer anwendungsspezifischen verschlüsselten Form in den entsprechenden Konfigurationsdateien abgelegt werden – der Datenbankzugriff selbst kann dann trotzdem über etablierte und bewährte Standardprotokolle erfolgen. Dadurch kann gleichzeitig dem Standardprinzip eines offenen Designs Rechnung getragen und dennoch der unkontrollierte Einsatz von Dritt-Software zumindest erschwert werden.

MULTI-FACTOR PROTECTION, LAYERED SECURITY

„Multi-Factor Protection" leitet sich vom Begriff der Multi-Faktor-Authentifizierung ab, wie man sie beispielsweise von Bankautomaten kennt: Zum Geldabheben werden sowohl die Bankkarte als auch die zugehörige PIN benötigt. Bringt ein Angreifer nur eines von beiden in seinen Besitz, so kann er damit noch nicht auf fremde Konten zugreifen. Die Grundidee bei Multi-Factor Protection ist, dass ein Sicherheitsbruch in jedem Fall mindestens zwei qualitativ verschiedene Angriffsaktionen erfordert, die für den Angreifer jeweils unabhängig voneinander mit entsprechendem Entdeckungsrisiko und Aufwand verbunden sind.

Ein einfaches Umsetzungsbeispiel des Multi-Factor-Protection-Prinzips wäre etwa der Transfer sensitiver Daten parallel über zwei (oder mehr) unabhängige Kanäle, wobei die Aufteilung so erfolgt (zum Beispiel durch ein Secret Sharing), dass das Abhören nur eines der Kanäle für einen Angreifer komplett nutzlos ist. Allerdings ist Multi-Factor Protection nur ein relativer Begriff: Wenn alle Kanäle aus diesem Beispiel bei einer gemeinsamen Empfangsstelle zusammenlaufen, wird dadurch zunächst lediglich Multi-Factor Protection gegen externe Angreifer erreicht. Zusätzliche Maßnahmen sind gegebenenfalls innerhalb der Empfangsstelle notwendig, um auch Multi-Factor Protection gegen mögliche interne Angreifer zu erhalten.

„Layered Security" kann als eine gewisse Abschwächung des Multi-Factor-Protection-Begriffs betrachtet werden und ist weitläufige Praxis in der IT-Sicherheit. Während bei Multi-Factor Protection gleichberechtigte Maßnahmen nebeneinander stehen und nur bei gleichzeitigem Aushebeln aller Maßnahmen ein Schadensfall möglich ist, besteht bei Layered Security eine Hierarchie an Schutzschichten. Mit jedem Bruch einer Schicht kann ein Angreifer etwas mehr Schaden anrichten, bis er auf die nächsttiefere Schicht stößt. Aufwand und Entdeckungsrisiko des Angreifers skalieren dadurch mit dem potentiellen Angriffsschaden.

Die Abgrenzung zwischen Multi-Factor Protection und Layered Security ist fließend und oft eine Frage der Sichtweise. Ein gängiges Praxisbeispiel für Layered Security ist eine Kombination von Maßnahmen, die für eine Verletzung der Vertraulichkeit sensitiver Daten alle gemeinsam ausgehebelt werden müssten, von denen aber nur eine die funktionale Verfügbarkeit des Systems schützt. Konkret könnte etwa der Zugriff auf einen Systemmonitor, über den die Anwendung auch heruntergefahren werden kann, über eine einfache Passwortabfrage gesichert sein, während die verarbeiteten Daten zusätzlich durch eine Verschlüsselung geschützt sind. Aus reiner Datenschutzsicht könnte man in solch einer Situation durchaus von Multi-Factor Protection sprechen.

Nichtsdestotrotz sollten beide Begriffe weniger als verschiedenrangige Ausprägung derselben Intention sondern vielmehr als gegenseitige Ergänzung aufgefasst werden. Multi-Factor Protection bezieht sich in erster Linie auf sehr spezifische Schutzziele (zum Beispiel die Wahrung der Vertraulichkeit ganz bestimmter Daten). Layered Security dagegen beschreibt eher eine globale Systemeigenschaft, die sämtliche relevanten Schutzziele umfasst. Dabei können durchaus alle Ebenen eines Layered-Security-Konsepts wieder über Multi-Factor-Mechanismen abgesichert sein, welche aber für unterschiedlich wichtige Schutzziele oder gegen unterschiedliche Angreifertypen unterschiedlich aufwändig und dementsprechend unterschiedlich stark ausfallen.

OFFLINE-SICHERHEIT

Zumindest der persistente Speicher jedes Systembereichs darf, wenn er vollständig in die Hände eines fähigen Angreifers gelangt, keine relevante Information preisgeben. Insbesondere dürfen keine kryptographischen Schlüssel im selben Systembereich abgespeichert sein wie damit verschlüsselte sensible Daten. Dadurch sollen Angriffe ausgeschlossen werden, bei denen ein Angreifer Daten aus dem System abgreift, während es abgeschaltet ist. Das betrifft insbesondere Angriffsszenarien, in denen der Angreifer gar keinen Zugriff auf lauffähige Systemteile hat, sondern beispielsweise nur auf ausrangierte Datenträger, die nicht ordnungsgemäß vernichtet wurden. Ebenso sollen aber auch Angriffe auf die Datenvertraulichkeit im laufenden System zumindest dadurch erschwert werden, dass der Angreifer den volatilen Arbeitsspeicher des Systems auslesen muss, was in aller Regel schwieriger ist, als an Festplatteninhalte zu gelangen. Lösungen, bei denen sämtliche Information im persistenten Speicher lokal vorhanden und lediglich mit Verschleierungstechniken (Code-Obfuscation, White-Box-Cryptography oder Ähnlichem) geschützt ist, können dieses Prinzip per Definition nicht erfüllen.

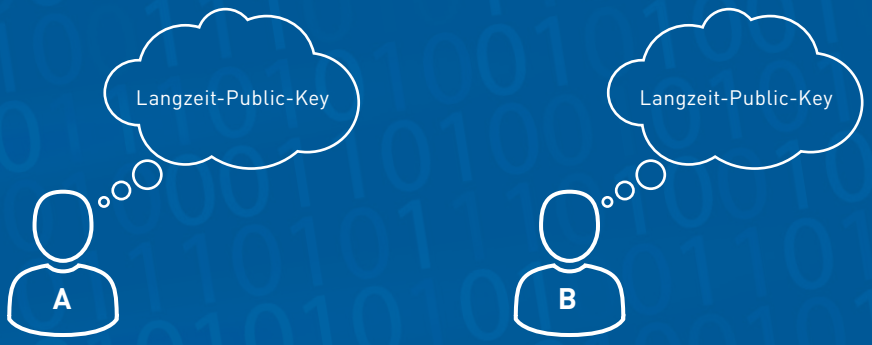
Folgende Aspekte, von denen in der Software-Architektur gerne abstrahiert wird, können ansonsten vorhandene Offline-Sicherheit komplett zunichte machen:

- Log-Files, in denen Daten persistiert werden, die laut Design eigentlich volatil sein sollten. Umfassendes Logging ist ein unverzichtbares Werkzeug zur Wartung und Analyse im Fehlerfall. Es muss aber grundsätzlich besondere Sorge getragen werden, welche Informationen mitgeloggt werden und welche nicht. Das gilt für alle verfügbaren Logging-Level-Konfigurationen, denn prinzipiell sollte Datenschutz Vorrang vor Bedienungskomfort haben.

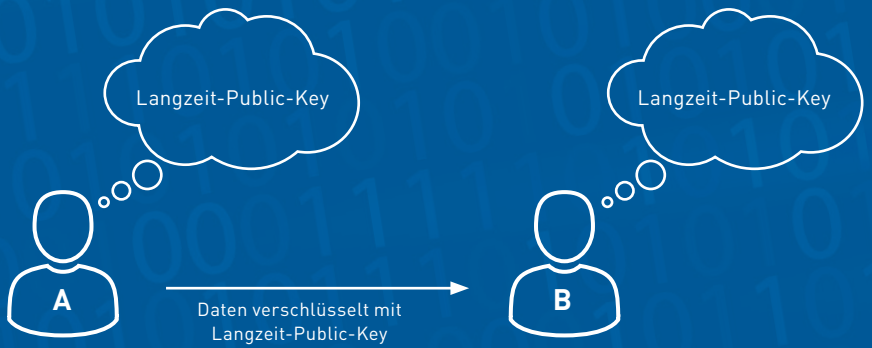
- Swap-Space, in den das Betriebssystem Daten temporär auslagert, wenn der volatile Arbeitsspeicher dafür nicht mehr ausreicht. Kann oder will man auf dieses Feature nicht komplett verzichten, so ist volltransparente Festplattenverschlüsselung ein möglicher Lösungsansatz, um Offline-Sicherheit trotzdem zu gewährleisten.

Abbildung: Beispielprotokolle für unidirektionalen Datentransfer mit und ohne Forward Secrecy.
Für bidirektionale Kommunikation können zwei solche unidirektionalen Lösungen in einander entgegengesetzter Richtung betrieben werden.

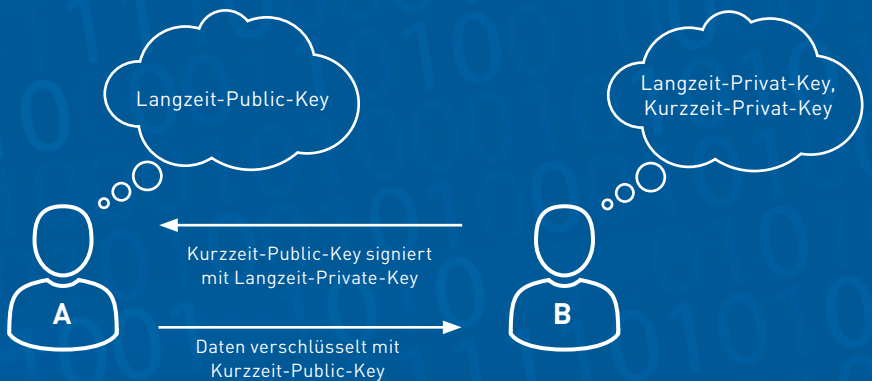
BEISPIELPROTOKOLLE FÜR UNIDIREKTIONALEN DATENTRANSFER MIT UND OHNE FORWARD SECRECY



Initiales Setup
Beide Parteien haben auf vertrauenswürdigem Weg ein Schlüsselpaar ausgetauscht (z.B. bei einem persönlichen Treffen oder über x.509-Zertifikate).



Datentransfer ohne Forward Secrecy
Wenn ein Angreifer den Datentransfer belauscht und beliebig später an den Langzeit-Private-Key gelangt, kann er damit im Nachhinein die Vertraulichkeit brechen.



Datentransfer mit Forward Secrecy
Langzeitschlüssel werden nur für Signaturen verwendet. Für die Verschlüsselung des eigentlichen Datentransfers wird ein kurzlebiges Sitzungsschlüsselpaar verwendet, welches überhaupt nicht persistent gespeichert wird. Die Signatur bei der Übertragung des Kurzzeit-Public-Key ist notwendig, damit kein Angreifer diesen durch einen eigenen Schlüssel ersetzen kann („Man-in-the-Middle-Angriff").

- Eine systemübergreifende Datensicherungsinfrastruktur, in der zentralisiert Backup-Kopien abgelegt werden. Daten, die laut Design eigentlich ausschließlich getrennt vorgehalten werden sollten, können so doch wieder an einem gemeinsamen Ort landen. Eine besonders dramatische Situation dieser Art liegt vor, wenn für verschiedene Systembereiche (mit jeweils für sich unkritischen Daten) unabhängig voneinander entschieden wird, eine öffentliche Cloud als Archiv zu nutzen.

FORWARD SECRECY

Im ursprünglichen Definitionssinn ist Forward Secrecy eine Sicherheitseigenschaft von Verschlüsselungsprotokollen: Wenn ein Angreifer verschlüsselte Kommunikation mitschneidet und anschließend an die jeweiligen privaten Langzeitschlüssel der beteiligten Parteien gelangt, darf es ihm dennoch nicht gelingen, damit die abgehörten Nachrichten im Nachhinein zu entschlüsseln. Nur wenn er bereits von Beginn an die Langzeitschlüssel besitzt und aktiv in den Nachrichtenaustausch eingreift, darf ein Bruch der Vertraulichkeit möglich sein. Gängige Verschlüsselungsprotokolle wie TLS erreichen diese Sicherheitseigenschaft, indem Langzeitschlüssel niemals direkt für die Nachrichtenverschlüsselung selbst verwendet werden, sondern nur um zwischen den beteiligten Parteien frische Sitzungsschlüssel signiert auszutauschen, welche nach Beendigung der Kommunikation umgehend wieder gelöscht werden.

Forward Secrecy als allgemeiner gefasstes Design-Prinzip bedeutet, dass Zeitfenster für eventuelle Angriffe möglichst eng begrenzt und A-posteriori-Angriffe ausgeschlossen sind. Werden beispielsweise Personendatensätze statistisch ausgewertet, die in ihrer Eingangsform notwendigerweise auch den bürgerlichen Namen im Klartext enthalten, so ist an dieser Stelle eine Privatsphärenverletzung durch einen internen Angreifer grundsätzlich möglich. Indem Klartextnamen vor dem Abspeichern mit einem entsprechend kurzlebigen Schlüssel verschlüsselt oder durch kryptographische Hashwerte ersetzt werden, kann man jedoch verhindern, dass einmal erfasste und abgespeicherte Datensätze später wieder über den bürgerlichen Namen real existierenden Personen zugeordnet werden.

HOMOGENITÄT / UNIFORMITÄT

Es lohnt sich, Sicherheitsmechanismen so zu gestalten, dass sie möglichst universell einsetzbar sind, und das Gesamtsystem einheitlich damit zu schützen. Ganz allgemein gilt bei Fragen der Sicherheit die Regel, dass jede Kette nur so stark ist wie ihr schwächstes Glied. Eine homogene Sicherheitslösung ist deshalb einem Flickenteppich aus Ad-hoc-Schutzmechanismen grundsätzlich vorzuziehen. Selbstredend darf dabei die universelle Einsetzbarkeit eines Sicherheitsmechanismus nicht erst durch komplexe Konfigurierbarkeit künstlich zustande kommen, denn damit geht die Geradlinigkeit verloren, die Nachvollziehbarkeit sinkt und es besteht eine erhöhte Gefahr für Implementierungs- und Anwendungsfehler.

Nur ein uniformes Sicherheitskonzept ermöglicht umfassenden Datenschutz bei vergleichsweise geringer Gesamtkomplexität und entsprechend niedriger Fehleranfälligkeit. Dazu gehört auch, dass ein lückenhafter Mechanismus nicht durch den Anbau weiterer Mechanismen gepatcht, sondern durch einen umfassenderen Mechanismus ersetzt wird. Im Fall einer Sicherheitslücke wegen eines Implementierungsfehlers wird diese Ersetzung in aller Regel durch einen entsprechenden Bugfix möglich sein. Größere Umgestaltungen sind bei konzeptionellen Schutzlücken notwendig; sei es, dass von Beginn an ein Designfehler vorlag oder dass nach einer funktionalen Anwendungserweiterung eine vormals hinreichende Sicherheitsmaßnahme nicht mehr alle potentiellen Angriffspunkte abdeckt.

Die Gesamtkomplexität und Fehleranfälligkeit eines Systems durch ein homogenes/uniformes Design möglichst gering zu halten, ist sicher auch ein sinnvolles Vorgehen in Projekten ohne direkten Bezug zu Privatsphärenschutz und/oder Big Data. Es erhält hier jedoch eine besondere Bedeutung, da eben jene Themen bereits inhärent eine hohe Komplexität mit sich bringen. Einfache Beispiele für besonders direkte Verletzungen dieses Prinzips sind ein redundanter Technologie-Stack (zum Beispiel durch Einbinden unterschiedlicher Libraries, mit denen jeweils dieselben verschlüsselten Daten entschlüsselt werden) oder der systeminterne Einsatz von unnötig vielen verschiedenen Datenformaten.

1.3 NOTWENDIGKEIT, STÄRKEN UND SCHWÄCHEN FORMALER ANONYMITÄTSBEGRIFFE

» Ein gut gegen Angriffe abgesicherter Berechnungsprozess garantiert noch nicht, dass die Berechnungsergebnisse datenschutzrechtlich unbedenklich sind. Um die Datenschutzkonformität gewonnener Statistiken sicherzustellen, ist als Grundlage ein geeigneter Anonymitätsbegriff erforderlich. Rein syntaktische Begriffe wie *k*-Anonymität sind zunächst recht einfach anzuwenden, haben jedoch schwerwiegende konzeptionelle Schwächen, die durch Zusatzmaßnahmen abgefangen werden müssen. Semantische Anonymität bietet starke und umfassende Sicherheitsgarantien, ist aber oft technisch schwierig zu realisieren. Umfassende Expertise ist zur korrekten Umsetzung in beiden Fällen unabdingbar.

Um aus Personendaten gewonnene Statistiken ohne Verletzung der geltenden Datenschutzbestimmungen verwerten zu können, muss unbedingt sichergestellt sein, dass sie tatsächlich keinen Personenbezug mehr enthalten. Die in den vorangegangenen Kapiteln vorgestellten Design-Prinzipien und kryptographischen Methoden dienen lediglich der Absicherung des Berechnungsprozesses an sich und liefern noch keinerlei Garantien bezüglich der Datenschutzkonformität des Berechnungsergebnisses. Dazu ist bei Weitem nicht hinreichend, Identifikatoren mit direktem Bezug zu einzelnen Individuen (wie zum Beispiel User-Namen in einem Online-Forum oder Sozialversicherungsnummern) aus der Datengrundlage zu entfernen. So konnten beispielsweise von AOL anonymisiert veröffentlichte Suchstatistiken vereinzelt wieder Personen zugeordnet werden.[1]

Auch die Kombination von einzeln möglicherweise unkritischen Attributen (zum Beispiel Geburtsjahr, Geschlecht und Postleitzahl des Wohnorts) kann für einen eindeutigen Personenbezug oft bereits genügen. Unter Umständen lässt sich sogar aus Daten ohne irgendwelche demographischen Attribute privatsphärenkritische Information ableiten und wieder individuellen Personen zuordnen – dies konnte anhand von Filmbewertungen demonstriert werden, die Netflix anonymisiert veröffentlicht hatte.[2]

Beide genannten Beispiele hatten für das jeweils beteiligte Unternehmen schwerwiegende juristische Konsequenzen und zeigen, dass für datenschutzkonforme Personendatenanalyse ein geeigneter Anonymitätsbegriff als Grundlage unbedingt erforderlich ist.

SYNTAKTISCHE ANONYMITÄTS-BEGRIFFE UND PRIVACY-PRESERVING DATA PUBLISHING

Der konzeptionell einfachste und in der Anwendung vermutlich am weitesten verbreitete Anonymitätsbegriff ist „*k*-Anonymity". Dabei wird gefordert, dass eine Information nur genau dann verwendet werden darf, wenn sie auf mindestens *k* Individuen zutrifft und somit keinen Einzelpersonenbezug hat. Dahinter steht ein Modell, bei dem Auszüge einer Datenbank in Tabellenform veröffentlicht werden und jede Tabellenzeile den Datensatz zu einem Individuum enthält. Der *k*-Anonymity-Begriff wird von solch einer Tabelle syntaktisch erfüllt, wenn jede ihrer Zeilen mindestens *k*-mal identisch vorkommt.

Diese Form wird typischerweise über „Generalization" und „Suppression" erreicht. Das heißt, dass zum einen Daten vergröbert werden (zum Beispiel, indem man bei Postleitzahlen die letzten Stellen ausblendet) und zum anderen manche Zeilen komplett weggelassen werden. Es gibt eine Vielzahl an Algorithmen, um über möglichst optimale Generalization- und Suppression-Strategien *k*-Anonymity zu erreichen und gleichzeitig eine möglichst hohe Aussagekraft der anonymisierten Daten zu erhalten.

Der *k*-Anonymity-Begriff bringt allerdings einige prinzipielle Schwächen mit sich, die den gewährleisteten Privatsphärenschutz stark relativieren:

- Sobald eine Gruppe von Individuen mit derselben Attributkombination hinreichend groß ist, erscheint sie als solche unverändert im anonymisierten Ergebnis. Das kann für einen Angreifer mit Kontextwissen genügen,

1 Michael Barbaro, Tom Zeller Jr.: *A Face Is Exposed for AOL Searcher No. 4417749*, The New York Times, 9. August 2006).
2 Arvind Narayanan, Vitaly Shmatikov: *Robust De-anonymization of Large Sparse Datasets*, IEEE Symposium on Security and Privacy 2008).

um datenschutzkritische Information zu rekonstruieren. Wenn beispielsweise in einer medizinischen Datenbank alle Patienten aus einem bestimmten geographischen Gebiet an derselben Krankheit leiden, kann das gegebenenfalls auch in der anonymisierten Version der Datenbank noch erkennbar sein. Zusammen mit der Kontextinformation, dass eine bestimmte Person in genau diesem Gebiet wohnt und an der Studie zur Erstellung der Datenbank teilgenommen hat, lässt sich dann direkt auf sensible medizinische Information über diese Person schließen.

- *k*-Anonymity ist nicht abgeschlossen unter paralleler Komposition. Das heißt, dass bei Verfügbarkeit mehrerer unterschiedlich anonymisierter Versionen derselben Datenbank keinerlei Anonymitätsgarantie mehr gewährleistet ist. Im Extremfall genügen bereits zwei solche unterschiedlich anonymisierte Versionen, um für einzelne Individuen sämtliche Attribute zu rekonstruieren. Damit ist *k*-Anonymity insbesondere für solche Szenarien ein formal unzureichender Begriff, in denen interaktive Anfragen an die zu anonymisierende Datenbank möglich sind und über Anfrageparameter das Anonymisierungsergebnis beeinflusst werden kann.

- Der *k*-Anonymity-Begriff bezieht sich formal ausschließlich auf das Anonymisierungsergebnis, ohne den verwendeten Anonymisierungsalgorithmus mit einzubeziehen. Rein formal produziert sogar ein Algorithmus, der einfach alle Datensätze ver-*k*-facht, ein *k*-anonymes Ergebnis. Solch offensichtliche Datenschutzlücken lassen sich noch recht einfach ausschließen, indem man fordert, dass Daten lediglich generalisiert und/oder komplett unterdrückt werden dürfen. Aber auch mit solchen Einschränkungen liefert *k*-Anonymity ganz generell keinerlei Garantien bezüglich der Information, die man potentiell aus Wissen über die Arbeitsweise des verwendeten Algorithmus zusätzlich zum eigentlichen Anonymisierungsergebnis gewinnen kann.

Alle genannten Schwächen basieren im Wesentlichen darauf, dass *k*-Anonymity ein rein syntaktischer Begriff ist, der isoliert das Ergebnis eines einzelnen Anonymisierungsvorgangs betrachtet. Damit ist dieser Begriff höchstens noch für Anwendungsgebiete geeignet, bei denen einmalig ein anonymisierter Komplettabzug einer Datenbank veröffentlicht werden soll („Privacy-Preserving Data Publishing") und möglichen Angreifern kein relevantes Kontextwissen aus Sekundärquellen zur Verfügung steht. Unabhängig davon ist bei *k*-Anonymity-basierten Lösungen auf jeden Fall darauf zu achten, welche Information durch die Arbeitsweise des verwendeten Anonymisierungsalgorithmus implizit zusammen mit dem eigentlichen Anonymisierungsergebnis veröffentlicht wird.

Es existieren Verschärfungen des *k*-Anonymity-Begriffs, die zusätzlich fordern, dass die Datengranularität beim Anonymisierungsvorgang um ein gewisses Mindestmaß reduziert wird (*l*-Diversity) und/oder dass Individuen zu Gruppen mit ähnlicher Struktur wie die Gesamtpopulation zusammen gefasst werden (*t*-Closeness), damit aus einer Gruppenzugehörigkeit nicht mehr auf individuelle Merkmale zurückgeschlossen werden kann. Da es sich dabei jedoch ebenso um rein syntaktische Begriffe handelt, die jeweils isoliert das Ergebnis eines einzelnen Anonymisierungsvorgangs betrachten, teilen sie letztendlich auch die entsprechenden prinzipiellen Schwächen von *k*-Anonymity.

Trotz dieser Probleme kann ein syntaktischer Anonymitätsbegriff je nach Einsatzgebiet dennoch ein geeignetes Mittel sein. Allerdings müssen dann im Design des Gesamtsystems und insbesondere bei der Auswahl bzw. Entwicklung des Anonymisierungsalgorithmus Zusatzvorkehrungen getroffen werden, um potentielle Datenschutzlücken auszuschließen, die der gewählte Anonymitätsbegriff formal nicht abdeckt. Wie bei allen Sicherheitsthemen besteht dabei ein erhebliches Potential für subtile, aber datenschutzkritische Fehler, dem nur mit entsprechender Expertise begegnet werden kann.

Es existieren auch noch weitere syntaktische Anonymitätsbegriffe, mit denen sich die hier genannten Schwächen zumindest in einem gewissen Rahmen umgehen lassen. Zum Beispiel bezieht „δ-Presence" explizit mögliches Kontextwissen eines potentiellen Angreifers mit ein – die entsprechenden Datenschutzgarantien hängen dann davon ab, wie akkurat sich dieses Kontextwissen im Vorfeld abschätzen lässt. In eine andere Richtung zielt der Begriff der „*m*-Invariance", welcher es in begrenztem Umfang

Abbildung: Beispiel für mögliche *k*-Anonymisierungen mit $k = 3$. Mittels entsprechendem Kontextwissen lassen sich bereits aus zwei unterschiedlichen Anonymisierungsergebnissen die Ursprungsdaten komplett rekonstruieren.

K-ANONYMISIERUNG MIT $K = 3$

W	80538
W	80538
W	80539
W	80539
M	80538
M	80538
M	80538
M	80539

Originaldaten
Jede Zeile entspricht einem Individuum und beinhaltet das Geschlecht und die Postleitzahl des Wohnortes.

W	8053*
W	8053*
W	8053*
W	8053*
M	80538
M	80538
M	80538

***k*-anonyme Variante I**
(letzte Zeile wurde unterdrückt)

W	8053*
W	8053*
W	8053*
W	8053*
M	8053*
M	8053*
M	8053*
M	8053*

***k*-anonyme Variante II**
(nur Generalisierung, keine Suppression)

*	80538
*	80538
*	80539
*	80539
*	80538
*	80538
*	80538
*	80539

***k*-anonyme Variante III**
(nur Generalisierung, keine Suppression)

Deanonymisierungsangriffe
Werden Variante I und Variante II zusammen veröffentlicht, so kann man aus dem Wissen, dass die Postleitzahlen 80530 bis 80537 real nicht existieren (und dem Wissen, dass der Anonymisierungsalgorithmus nicht unnötig Zeilen wegwirft), bereits auf folgende Datenstruktur rückschließen:

W	8053*
W	8053*
W	8053*
W	8053*
M	80538
M	80538
M	80538
M	80539

Weiß man ferner über die Arbeitsweise des Algorithmus, dass die Abwägung zwischen Generalisierung und Supression in sich konsistent ist (also z.B. die Frage: „Soll ich vier Einträge um eine Stufe generalisieren oder lieber eine Zeile unterdrücken?" für den "weiblichen" und den „männlichen" Teil der Tabelle nie unterschiedlich entschieden würde), so kann man weiter schlussfolgern, dass der „weibliche" Teil der Tabelle aus zwei Zweiergruppen bestehen muss. Damit sind aus Variante I und Variante II die Urspungsdaten vollständig rekonstruierbar.

erlaubt, verschiedene anonymisierte Versionen einer über die Zeit veränderlichen Datenbank zu veröffentlichen.

SEMANTISCHE ANONYMITÄT UND PRIVACY-PRESERVING DATA MINING

Eine alternativer Begriff, welcher viele der wesentlichen Schwächen rein syntaktischer Anonymitätsdefinitionen umgeht, ist „Differential Privacy". Dabei wird gefordert, dass der Anonymisierungsprozess randomisiert ist und jede Änderung an einem einzelnen Datensatz der Ursprungsdaten nur unwesentlichen Einfluss auf die entsprechende Ausgabeverteilung hat. Dadurch ist sichergestellt, dass die Daten eines einzelnen Individuums nicht im Anonymisierungsergebnis wiederzuerkennen sind (denn sonst würde die Änderung dieser Daten zu einer signifikanten Änderung in der Ausgabe des Anonymisierungsalgorithmus führen) und zwar unabhängig von etwaigem Kontextwissen.

Durch den direkten Bezug des Anonymitätsbegriffs auf den verwendeten Algorithmus sind auch entsprechende Seitenkanäle ausgeschlossen, durch die ein Angreifer aus dem Wissen über die Arbeitsweise des Verfahrens auf implizite Zusatzinformation rückschließen könnte. Da der Begriff außerdem rein auf Abstandsmaßen zwischen Wahrscheinlichkeitsverteilungen basiert, ist er komplett unabhängig von der verwendeten Ausgabesyntax. Anders als bei den vorgestellten syntaktischen Begriffen besteht insbesondere keine Beschränkung auf tabellenartige Ergebnisformate.

Ferner kann aus den zugrundeliegenden Abstandsmaßen ein Kompositionstheorem abgeleitet werden, welches die verbleibenden Anonymitätsgarantien bei Veröffentlichung unterschiedlicher anonymisierter Datenversionen exakt quantifiziert. Dabei ist unerheblich, ob genau dieselben Daten auf verschiedene Weise anonymisiert wurden oder ob sich die Ursprungsdaten zwischenzeitlich geändert haben. Diese Eigenschaft macht Differential Privacy zu einem recht natürlichen Begriff für Anwendungsfälle, bei denen Antworten auf komplexe interaktive Datenbankanfragen on-demand anonymisiert werden sollen („Privacy-Preserving Data Mining") und/oder die Datenbasis zeitlichen Veränderungen unterworfen ist. Aber auch in rein statischen Anwendungsfällen bieten bereits die Garantie auf Seitenkanalfreiheit und die Robustheit gegen alle Arten von kontextbasierten Deanonymisierungsangriffen einen nicht zu unterschätzenden Mehrwert.

Den offenkundigen Vorteilen des Differential-Privacy-Begriffs stehen allerdings auch einige Nachteile gegenüber. So ist die mathematische Definition wesentlich weniger intuitiv als etwa bei *k*-Anonymity. Entsprechend schwierig gestaltet sich die Kommunikation, wenn die Sicherheit des Verfahrens gegenüber Laien (zum Beispiel von der Datenanalyse betroffenen Nutzern) dargelegt werden soll. Ferner ist der Begriff parametrisiert und die Güte der gewährleisteten Datenschutzgarantien hängt vollständig von einer hinreichend restriktiven Parameterwahl ab. Übertrieben restriktive Parameter sind allerdings ebenfalls kontraproduktiv, denn bei steigendem Anonymitätsniveau sinkt zwangsläufig die Datenqualität der Anonymisierungsergebnisse. Für die richtige anwendungsspezifische Abwägung zwischen Datennutzen und Anonymitätsniveau ist aufgrund der mathematisch anspruchssvollen Begriffsdefinition Expertenwissen auch auf Seiten der entsprechenden Entscheidungsträger notwendig. Außerdem sind selbst bei Differential Privacy der Komponierbarkeit von verschiedenen anonymisierten Datenanalysen gewisse Grenzen gesetzt – es ist ganz prinzipiell unmöglich, Datennutzung in beliebigem Umfang zuzulassen, ohne dadurch letztendlich die Privatsphäre komplett aufzuheben.

Nicht zuletzt gestaltet sich die algorithmische Umsetzung eines Differential-Privacy-Mechanismus zuweilen als äußerst schwierig. Oft müssen ausgefeilte Speziallösungen entwickelt werden, denn die existierenden generische Algorithmen sind entweder vom Rechenaufwand her nicht praktikabel oder nur für spezielle Anwendungsfälle geeignet. Trotz all dem sollte der Rückgriff auf Differential Privacy als formale Grundlage für datenschutzkonforme Personendatenanalyse grundsätzlich eine Überlegung wert sein. Schließlich führt im Sinne einer nachhaltigen Entwicklungsstrategie eigentlich kein Weg an einem konzeptionell möglichst starken Anonymitätsbegriff vorbei.

2 VERANKERUNG VON DATENSCHUTZ IN DER SOFTWARE-ENTWICKLUNG

» In der modernen, agilen Software-Entwicklung muss Datenschutz sinnvoll in den Entwicklungsprozess integriert werden. Nicht nur bei der Entwicklung der Software, auch bei der Architektur ist der Datenschutz zu berücksichtigen. Methoden, die im Rahmen der professionellen Software-Entwicklung eingesetzt werden, bieten auch gute Ansätze für die Integration von IT-Security in den Entwicklungsprozess.

Datenschutz ist auch bei Beachtung der bis hierhin vorgestellten Prinzipien keine einfache Aufgabe. Der Entwurf eines System unter Berücksichtigung der hier beschriebenen Design-Prinzipien ist immer nur der erste Schritt. Alle wichtigen und produktiv genutzten Software-Systeme werden in der Realität gewartet, um zusätzliche Features erweitert, oft auch grundlegend umgebaut – und stets muss dabei der Datenschutz geprüft werden.

In einem klassischen Projektumfeld, in dem neu entwickelte Software wenige Male im Jahr ausgeliefert wird, mag es ausreichend sein, vor jedem Release die Software auf Verletzungen von Datenschutz zu prüfen – etwa durch externe Audits oder Penetrationstests. Doch unter dem Druck der „Digitalen Transformation" geht die Tendenz in der Software-Entwicklung zu mehr Flexibilität, sodass zum Beispiel Features im Wochenrhytmus oder sogar noch schneller agil geliefert werden. Bei Amazon etwa wird im Schnitt alle 11,6 Sekunden ein Deployment durchgeführt. Der große Vorteil und ein wichtiger Grund für die Verbreitung agiler Methoden in der Software-Entwicklung ist der Gewinn an Flexibilität. Gerade bei sich schnell ändernden rechtlichen Anforderungen und notwendigen Verbesserungen der Datensicherheit kann der Wert dieser Flexibiltät nicht genug betont werden. Allerdings muss es ein Konzept geben, den Schutz sensibler Daten in diesem agilen Umfeld sicherzustellen.

Wenn man sich mit agilen Methoden beschäftigt, ist eines der prägenden Merkmale der Aufbau cross-funktionaler Teams. Dieses Prinzip wirkt darauf hin, in einem Team möglichst viel verschiedenes Wissen bzw. die entsprechenden Personen zusammenzubringen. Zuerst betraf dies vor allem die Bereiche Entwicklung und Test, aktuell wird unter dem Stichwort DevOps auch ein Fokus auf operative Aufgaben gelegt, die vom Entwicklungsteam übernommen werden können. Und auch die Entwicklung eines Software-Architektur-Konzepts ist eine Aufgabe, die im Rahmen von agiler Software-Entwicklung vom ganzen Team getragen wird. Bei all diesen Aufgaben bringen verschiedene Mitglieder des agilen Teams unterschiedliche Grade an Erfahrung bzw. Vorwissen und Spezialisierung mit. So sind oft die meisten Teammitglieder erfahrene Software-Entwickler, während einige einen starken Operations- bzw. Architektur-Hintergrund haben. Diese einzelnen Personen übernehmen dann auch einen großen Teil der anfallenden Aufgaben in den Bereichen, aber verbreiten gleichzeitig, etwa durch Pairing, das Wissen und die Erfahrung im ganzen Team.

Ebenso kann man Security in einem agilen Umfeld leben: Jedes Team sollte einen „Security Champion" haben, der sensibilisiert und erfahren ist im Umgang mit Datenschutz-Themen und diese Erfahrung und Sensibilisierung kontinuierlich im gesamten Team verbreitet. Der große Vorteil bei diesem Vorgehen liegt darin, dass der Datenschutz so zu einem Prinzip wird, welches vom ganzen Team gelebt wird, und zu einem viel früheren Zeitpunkt und mit einer viel größeren Abdeckung stattfindet. Im Gegensatz dazu kann ein nicht im Team integrierter Security-„Aufseher" dazu führen, dass Datenschutz-Themen nicht als Aufgabe des Teams gesehen werden und zuwenig in die Entwicklung einfließen.

Ein nicht zu vernachlässigender Punkt ist allerdings, dass gerade Security-Experten längst nicht so einfach zu finden oder auszubilden sind wie Architekten, System-Administratoren oder Tester. Bei großen Projekten sollte man trotzdem der Versuchung widerstehen, nur einen Experten für viele Teams zu haben oder stattdessen ein einziges, sehr großes Team zu bilden. Hier kann es sinnvoll sein, den Security- Experten durch die Teams rotieren zu lassen, aber auf jeden Fall pro Team eine Person zu

finden, die sich in die Datenschutz-Themen vertieft und sich etwa in einer Community of Practice mit den anderen Security Champions regelmäßig austauscht.

Ein weiterer wichtiger Aspekt ist auch die Architektur. Der notwendige Bedarf an Datenschutz muss sich unbedingt in den Qualitätszielen (oder nicht-funktionalen Anforderungen) des Projektes und dementsprechend in der Architektur widerspiegeln. Denn ein Fokus auf bestimmte Qualitätsziele führt oft dazu, dass andere Qualitätsziele schwerer zu erreichen sind. Wenn der Datenschutz eine große Rolle spielt, wird etwa das Qualitätsziel der Analysierbarkeit schwieriger zu erreichen – schließlich dürfen sensible Kundendaten, die eventuell zu Fehlern geführt haben, nicht einfach in Logfiles geschrieben werden. Auch Zeitverhalten und Ressourcenverbrauch stellen auf einmal eine größere Herausforderung dar, da kryptographische Operationen oft sehr ressourcenhungrig sind. Es werden also sicherlich je nach Projektkontext spezifische Ideen und Kompromisse notwendig sein.

Umso wichtiger werden in solchen Projekten die Prinzipien, die heutzutage zum Standard bei professioneller Software-Entwicklung gehören sollten: Automatisierte Tests und Continuous Deployment. Die Tests minimieren die Anzahl an Fehlern, die ins Produktivsystem kommen, das Continuous Deployment ermöglicht schnellste Reaktionszeiten, wenn etwa ein sicherheitskritischer Bug behoben werden muss. Die automatisierten Tests sollten selbstverständlich auch die sicherheitsrelevanten Features testen und sicherstellen, dass Änderungen, die den Schutz der Daten negativ beeinflussen könnten, gar nicht erst ins Produktivsystem kommen können. Gerade für bekannte sicherheitsrelevante Probleme gibt es Tool-Unterstützung, um Sicherheitsprobleme zu identifizieren. Systeme, die mit sensiblen Kundendaten arbeiten, sollten mit einem hohen Fokus auf Qualität hergestellt werden. Führt schlechte Software-Entwicklung bei anderen Systemen lediglich zu Verzögerungen bei Timelines und schlecht wartbarer Software, so können Fehler bei dieser Art von Software existenzbedrohliche Konsequenzen nach sich ziehen.

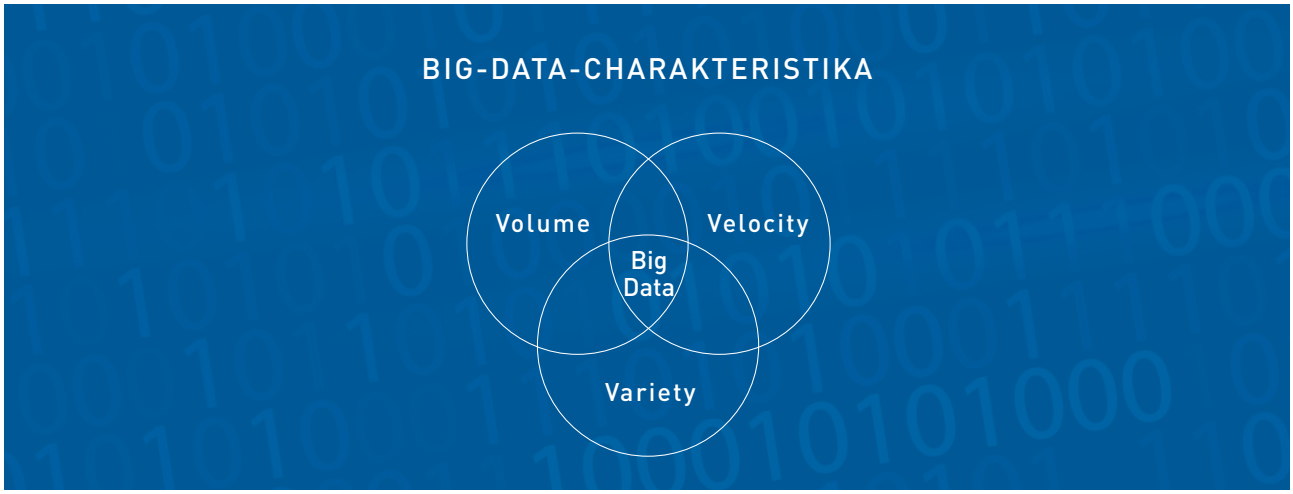
3 BIG DATA – TECHNISCHE HERAUSFORDERUNGEN UND LÖSUNGEN

» Große Datenmengen bergen das Potential für statistisch relevante und damit wertvolle Analysen. Außerdem sind einzelne Individuen in größeren Gruppen schwieriger zu identifizieren, was eine datenschutzkonforme Verarbeitung begünstigt bzw. überhaupt erst ermöglicht. Hohe Datenvolumina bringen aber auch technische Herausforderungen mit sich. Entsprechend sorgfältige Abwägung ist bei Entscheidungen zur Rechenumgebung und den eingesetzten Technologien notwendig, um den Big-Data-Charakteristika *Volume*, *Variety* und *Velocity* gerecht zu werden.

Moderne IT ermöglicht die Erfassung, Sammlung und Speicherung riesiger Datenmengen sowohl im Hinblick auf die unmittelbare Anwendung (z.B. Bilddatenverarbeitung in der Medizin) als auch im Sinne einer Zusatzfunktion neben der unmittelbaren Anwendung (z.B. Aufzeichnung von Systemdiagnosedaten oder des Kundenverhaltens in der Automobilelektronik oder in der Telekommunikation). Auf diese Weise entstehen sehr umfangreiche Datensätze, aus denen durch statistische Methoden wertvolle Erkenntnisse gewonnen werden können. Aus diesem Grund nehmen immer mehr Unternehmen neben ihrem traditionellen Kerngeschäftsmodell die in diesem Kontext gewonnenen Daten als wertvolle Ressource wahr.

Daraus resultieren zum einen eine Reihe von organisatorischen Themen innerhalb eines Unternehmens, die sich unter dem Stichwort „Data Governance" subsumieren lassen – d.h. es muss geklärt werden, wer für Datenaufzeichnung, Auswertung, Datenschutz und Auskunftsansprüche gegenüber Kunden verantwortlich ist, bzw. welche Richtlinien dafür beachtet oder neu geschaffen werden müssen. Andererseits ergeben sich auch technische Herausforderungen, insbesondere im Kontext der im oberen Teil geschilderten Datenschutzanforderungen. Prinzipiell kann die Beachtung des Datenschutzes bei der Datenverarbeitung im ersten Schritt durch die bloße Menge der Daten einfacher werden. Als konkretes

Abbildung: Big-Data-Charakteristika *Volume* (Datenmenge), *Variety* (Vielfalt der Datenquellen und -formate) und *Velocity* (Geschwindigkeit der Datenerhebung und des Datentransfers).



Beispiel: Soll von einzelnen Kunden auf größere, *k*-anonyme Gruppen abstrahiert werden, kann bei vorgegebenem *k* umso mehr Information erhalten werden, je mehr zu anonymisierende Kundendaten vorliegen. Andererseits kann bei größerer Kundenbasis der Parameter *k* auch höher gewählt werden, ohne dass die Aussagekraft der Analyse signifikant darunter leidet. Das einzelne Individuum ist dann als Teil der umfangreicheren Gruppe schwieriger identifizierbar und damit anonymer.

Allerdings wird Datenverarbeitung nicht alleine dadurch anonym und datenschutzkonform, dass das Datenvolumen besonders groß ist. Zudem führen große Datenvolumina auf technischer Seite teilweise zu Problemen – größere Datenmengen bedeuten automatisch mehr Verarbeitungsaufwand, wobei auch zu berücksichtigen ist, dass die im Umfeld von Anonymisierungen verwendeten kryptographischen Verfahren tendenziell hohe Rechenkomplexität haben. So kann die Verarbeitung großer Datenvolumina unter Berücksichtigung der notwendigen Anonymisierung schnell kostspielig werden, was jedoch kein valides Argument für die Reduktion von Datenschutzstandards sein darf.

Um das Problem der hohen Verarbeitungskosten zu lösen, bieten sich Parallelisierung und verteilte Berechnungen bei den gewünschten Auswertungen an. Auf der anderen Seite darf nicht vernachlässigt werden, dass die Entscheidung, Berechnungen in verteilten Systemen durchzuführen, zu einer deutlich gesteigerten Systemkomplexität führt. Prinzipiell sind die gängigen Werkzeuge zwar darauf ausgelegt, auch mit Ausfällen umgehen zu können – in der Praxis erfordert dies jedoch einen Zusatzaufwand, der bei rein serieller Ausführung vermieden wird.

Falls die Menge der zu prozessierenden Daten die Kapazitäten gängiger Hauptspeicher bei weitem übersteigt (*Volume*), steht ein großer Zoo an möglichen Technologien zu Verfügung, die darauf ausgelegt sind, viele und verteilte Ressourcen optimal auszulasten. So bietet sich die Verwendung einer Plattform zur Ressourcenverwaltung wie Apache Hadoop[1] oder Apache Mesos[2] an. Während Hadoop sich als Quasi-Standard im Umfeld Batch-Prozessierung und Data Lake herauskristallisiert hat, ist Mesos eine jüngere Technologie. Diese wird besonders dann interessant, wenn die zu verwaltenden Ressourcen nicht ausschließlich für Big-Data-Berechnungen genutzt werden – so lassen sich dort beispielsweise auch containerisierte Web Services provisionieren.

Grundsätzlich ist es ein nicht zu unterschätzender Vorteil, auf die Prozessierung inhärent heterogener Daten (*Variety*) mit spezifischen Lastprofilen vorbereitet zu sein. Darüber hinaus ist es empfehlenswert, zur Entwicklung der konkreten Analyse-Applikationen ein Framework mit stärkeren Abstraktionen zu verwenden – beispielsweise Apache Flink[3] oder Apache Spark[4]. Sowohl Flink als auch Spark setzen auf Hadoop/Mesos auf und bieten Entwicklern leicht zugängliche Grundfunktionalitäten sowie darauf aufbauende APIs – konkret etwa für Berechnungen auf Graphen, im Machine-Learning-Umfeld oder via SQL, was im Data-Science- und Business-Intelligence-Kontext geläufig ist.

Spielt zusätzlich die Anforderung von Echtzeitverarbeitung, das heißt niedriger Latenzen (*Velocity*), eine gesteigerte Rolle, lohnt sich die Evaluation eines für Streaming-Anwendungen ausgelegten Messaging-Systems wie Apache Kafka[5]. Beispielsweise in Kombination mit Flink lassen sich mit vertretbarem Aufwand Real-Time-Pipelines errichten, die robust gegen Ausfälle sind, mit fachlichen Herausforderungen wie Event-Time-Skew (in fachlich falscher Reihenfolge eintreffende Datenereignisse) gut umgehen können und dabei gleichzeitig höchsten Ansprüchen an Skalierbarkeit und Exaktheit im Sinne von Auslieferungsgarantien (Stichwort „exactly-once") genügen.

Eine maßgebliche Entscheidung ist die Auswahl der Umgebung, in der die Daten erhoben, gelagert und prozessiert werden. Auf einem Spektrum von On-Premise bis Cloud gibt es viele Möglichkeiten. An einem Ende bleiben alle sensiblen Daten innerhalb des Unternehmens, was im Sinne des Need-to-Know-Prinzips einen rationalen Default darstellt. Dem entgegen steht die Tatsache, dass ein Großteil der Hardware die meiste Zeit brachliegen würde – schließlich sollte sinnvollerweise sehr großzügig dimensioniert werden. Kapazitäten zusätzlich zu

den erwarteten Spitzenaufwänden erhöhen einerseits die Robustheit der Datenverarbeitung gegen einzelne Ausfälle und sind andererseits etwaigen nach Systemupdates anfallenden Nachverarbeitungen zuträglich.

Am anderen Ende des Spektrums liegen Cloud-Services – damit lassen sich Ressourcen flexibel je nach Anwendungsfall hinzukaufen und verursachen nach Benutzung keine weiteren Kosten. Die verschiedenen Cloud-Computing-Anbieter bieten elastische Skalierbarkeit – das heißt, im Idealfall kann das System sich selbstständig auf variierende Lasten einstellen. Dafür muss hier in noch höherem Maße ein kritisches Augenmerk auf die Anonymität der Daten gelegt werden. Die Technologieführer gehören zu US-Mutterkonzernen, sodass ein Zugriff von Seiten des Anbieters nicht ausgeschlossen werden kann (sowohl aus technischen Gründen als auch wegen nationaler Gesetzgebung wie etwa dem Patriot Act). Sind an datenschutzkritischen Verarbeitungen prinzipiell externe Parteien beteiligt, hat dies auch besondere juristische Konsequenzen (Stichwort „Auftragsdatenverarbeitung").

Abhängig von den Anforderungen eines konkreten Anwendungsfalls gilt es, Vor- und Nachteile von On-Premise-Lösungen, Public- oder Mixed-Cloud (Aufteilung des Systems auf verschiedene Umgebungen) abzuwägen. Selbstverständlich geht mit jeder dieser Entscheidungen wie auch bei der weiter oben angesprochenen Technologiewahl einher, dass entsprechende Kompetenzen im Unternehmen, falls noch nicht vorhanden, aufgebaut oder hinzugekauft werden müssen.

Referenzen

1 https://hadoop.apache.org
2 http://mesos.apache.org
3 https://flink.apache.org
4 http://spark.apache.org
5 https://kafka.apache.org

TNG TECHNOLOGY CONSULTING

Kontakt
TNG Technology Consulting GmbH
Betastraße 13a
85774 Unterföhring
Deutschland
Telefon: +49-89-2158996-0
Fax: +49-89-2158996-9
E-Mail: info@tngtech.com
www.tngtech.com